# XXI Autumn Meeting of Polish Information Processing Society

## Conference Proceedings


ISBN 83-922646-0-6

M. Ganzha, M. Paprzycki, J. Wachowicz, K. Węcel
(editors)

Wisła 2005

Typesetting: A. Denisjuk

# XXI Autumn Meeting of PIPS. Conference Proceedings

## TABLE OF CONTENTS

ISBN 83-922646-0-6


Dear Readers,

It is our pleasure to introduce you to this volume of proceedings of the Scientific Session organized within the framework of XXI Autumn Meeting of Polish Information Processing Society (Hotel Gołębiewski, Wisła, Poland, December 5-9, 2005). Autumn Meetings of Polish Information Processing Society are one of the longest existing Polish conferences devoted to all aspects of computer science and information technology. The first Meeting took place in 1984 and since then it has evolved through a number of forms. This year the Meeting consists of invited and sponsored lectures selected in such a way to attract IT professionals and managers working in all areas of industry (both "IT producers" and "IT users"). Invited lectures were supplemented by a number of workshops, panels and tutorials. Furthermore, for the first time in its history, a Scientific Special Session, consisting of refereed contributions was organized. The aim of this session was to present most interesting results obtained in broadly understood academic computer science and information technology.

Invited lectures were presented by Andrzej Śluzek from Nanyang Technological University in Singapore and Yaroslav Sergiejew from Uniwersity of Calabria in Rende. A total of 43 submissions have been received and after refereeing, 27 were accepted for presentation.

We would like to express our gratitude to PROKOM S.A. for being the sole sponsor of the Scientific Session. Furthermore, we are very grateful to the Program Committee, consisting of:

Witold Bielecki
Jacek Błażewicz
Mieczysław Budzyński
Włodzisław Duch
Jan Duda
Dariusz Dziuba
Roman Galar
Krzysztof Goczyła
Piotr Jędrzejowicz
Joanna Józefowska
Janusz Kacprzyk
Jerzy Kisielnicki
Józef Korbicz
Anna Kucaba-Piętal
Halina Kwaśnicka
Jacek Mańdziuk
Zygmunt Mazur
Edward Nawarecki
Ngoc Thanh Nguyen
Wojciech Olejniczak
Lech Polkowski
Roman Słowiński
Stanisław Stanek
Bogdan Stefanowicz
Piotr Szczepaniak
Tadeusz Szuba
Marek Tudruj
Maciej Wygralak

for their contributions to the success of the conference.

We hope that materials contained in this volume will satisfy your expectations and entice you to submit your own contributions next year, when we will be organizing the II Scientific Session within the framework of the XXII Autumn Meeting of Polish Information Processing Society.

Maria Ganzha
Marcin Paprzycki
Jacek Wachowicz
Krzysztof Węcel

ISBN 83-922646-0-6


Drodzy Czytelnicy!

Z przyjemnością oddajemy w Wasze ręce tom materiałów konferencyjnych z Sesji Naukowej organizowanej w ramach XXI Jesiennych Spotkań PTI (Hotel Gołębiewski, Wisła, 5-9 grudnia, 2005). Jesienne Spotkania PTI są jedną z najdłużej istniejących konferencji informatycznych w Polsce. Pierwsze spotkania odbyły się w roku 1984 i do dziś przeszły one przez wiele różnych faz i form. W roku bieżącym konferencja ta składała się z zaproszonych i sponsorowanych wykładów dobranych tak, aby zachęcić do udziału informatyków i menadżerów pracujących we wszystkich działach gospodarki. W programie konferencji znalazły się również warsztaty, panele i tutoriale. Równocześnie, po raz pierwszy w ramach Jesiennych Spotkań, zorganizowana została Sesja Naukowa składająca się z recenzowanych tekstów. Celem Sesji Naukowej była prezentacja tego, co najlepsze i najciekawsze w informatyce akademickiej.

Referaty zaproszone wygłoszone zostały przez Andrzeja Śluzka z Nanyang Technological University w Singapurze oraz Yarosława Sergiejewa z Uniwersytu Kalabrii w Rende. Na konferencje nadesłano 43 referaty, z których po recenzjach przyjęto 27.

Serdecznie dziękujemy firmie PROKOM S. A. za bycie wyłącznym sponsorem Sesji Naukowej. Równocześnie dziękujemy Komitetowi Programowemu konferencji w składzie:

Witold Bielecki
Jacek Błażewicz
Mieczysław Budzyński
Włodzisław Duch
Jan Duda
Dariusz Dziuba
Roman Galar
Krzysztof Goczyła
Piotr Jędrzejowicz
Joanna Józefowska
Janusz Kacprzyk
Jerzy Kisielnicki
Józef Korbicz
Anna Kucaba-Piętal
Halina Kwaśnicka
Jacek Mańdziuk
Zygmunt Mazur
Edward Nawarecki
Ngoc Thanh Nguyen
Wojciech Olejniczak
Lech Polkowski
Roman Słowiński
Stanisław Stanek
Bogdan Stefanowicz
Piotr Szczepaniak
Tadeusz Szuba
Marek Tudruj
Maciej Wygralak

za wkład w sukces konferencji.

Mamy nadzieję, że zawarte w materiałach teksty spełnią oczekiwania oraz skłonią do nadesłania propozycji prezentacji za rok, gdy będziemy organizować II Sesję Naukową w ramach XXII Spotkań Jesiennych PTI.

Maria Ganzha
Marcin Paprzycki
Jacek Wachowicz
Krzysztof Węcel

ISBN 83-922646-0-6


Дорогие читатели!

С радостью предоставляем вашему вниманию том материалов Научной сессии организованной в рамках конференции XXI Осенние встречи Польского Информатического Общества (Отель «Голембёвский», Висла, 5-9 декабря 2005). Осенние встречи – одна из старейших конференций в Польше – первые Встречи состоялись в 1984 году. За время своего существования конференция прошла через различные фазы и формы. С целью привлечь к участию в конференции как можно больше информатиков и менеджеров, работающих в различных сферах народного хозяйства, были организованы лекции и презентации, целью которых было привлечь к участию в конференции; организованы также различные семинары и круглые столы. Впервые в рамках конференции была организована Научная сессия, состоящая из рецензированных статей. Целью Научной сессии было представление того, что наиболее ценно и интересно в академической информатике.
Для участия в сессии были приглашены Анджей Шлюзка (Andrzej Śluzka) из Nanyang Technological University (Сингапур) и Ярослав Сергеев из Kalabria Uniwersity в Rende (Италия). На конференцию в общей сложности было прислано 43 статьи, из которых в результате рецензирования осталось только 27.

Сердечно благодарим фирму PROKOM S. A. за спонсирование Научной сессии.

Особую благодарность выражаем Программному Комитету конференции в составе:

| | | |
|---|---|---|
| Witold Bielecki | Janusz Kacprzyk | Lech Polkowski |
| Jacek Błażewicz | Jerzy Kisielnicki | Roman Słowiński |
| Mieczysław Budzyński | Józef Korbicz | Stanisław Stanek |
| Włodzisław Duch | Anna Kucaba-Piętal | Bogdan Stefanowicz |
| Jan Duda | Halina Kwaśnicka | Piotr Szczepaniak |
| Dariusz Dziuba | Jacek Mańdziuk | Tadeusz Szuba |
| Roman Galar | Zygmunt Mazur | Marek Tudruj |
| Krzysztof Goczyła | Edward Nawarecki | Maciej Wygralak |
| Piotr Jędrzejowicz | Ngoc Thanh Nguyen | |
| Joanna Józefowska | Wojciech Olejniczak | |

за их вклад в организацию Научной сессии.

Мы очень надеемся, что содержащиеся в материалах статьи заинтересуют вас и через год, когда будем организовывать II Научную сессию в рамках XXII Осенних встреч, вы пришлёте нам свои работы.

Maria Ganzha
Marcin Paprzycki
Jacek Wachowicz
Krzysztof Węcel

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 1–8 

# Mathematical Foundations of the Infinity Computer

Yaroslav D. Sergeyev

[1] Dipartimento di Elettronica, Informatica e Sistemistica,
Università della Calabria, Via P. Bucci, Cubo 41-C, 87030 Rende (CS), Italy
[2] Software Department,
N. I. Lobatchevsky State University, Nizhni Novgorod, Russia yaro@si.deis.unical.it
http://wwwinfo.deis.unical.it/∼yaro

**Abstract.** All the existing computers are able to execute arithmetical operations only with finite numerals. Operations with infinite and infinitesimal quantities could not be realized. The paper describes a new positional system with infinite radix allowing us to write down finite, infinite, and infinitesimal numbers as particular cases of a unique framework. The new approach both gives possibilities to execute calculations of a new type and simplifies fields of mathematics where usage of infinity and/or infinitesimals is required. Usage of the numeral system described in the paper gives possibility to introduce a new type of computer – Infinity Computer – able to operate not only with finite numbers but also with infinite and infinitesimal ones.

**Key Words**: Infinite and infinitesimal numbers, Infinity Computer, divergent series.

## 1 Introduction

Problems related to the idea of infinity are among the most fundamental and have attracted the attention of the most brilliant thinkers throughout the whole history of humanity. Numerous trials (see [2–6, 8, 10]) have been done in order to evolve existing numeral systems and to include infinite and infinitesimal numbers in them. To emphasize importance of the subject it is sufficient to mention that the Continuum Hypothesis related to infinity has been included by David Hilbert as the problem number one in his famous list of 23 unsolved mathematical problems that have influenced strongly development of Mathematics and Computer Science in the XXth century (see [6]).

The point of view on infinity accepted nowadays is based on the famous ideas of Georg Cantor (see [2]) who has shown that there exist infinite sets having different number of elements. However, it is well known that Cantor's approach leads to some paradoxes. The most famous and simple of them is, probably, Hilbert's paradox of the Grand Hotel (see, for example, [12]). Problems arise also in connection with the fact that usual arithmetical operations have been introduced for a finite number of operands.

There exist different ways to generalize traditional arithmetic for finite numbers to the case of infinite and infinitesimal numbers (see [1, 2, 4, 8, 10]). However, arithmetics developed for infinite numbers are quite different with respect to the finite arithmetic we are used to deal with. Moreover, very often they leave undetermined many operations where infinite numbers take part (for example, infinity minus infinity, infinity divided by infinity, sum of infinitely many items, etc.) or use representation of infinite numbers based on infinite sequences of finite numbers. These crucial difficulties did not allow people to construct computers that would be able to work with infinite and infinitesimal numbers in the same manner as we are used to do with finite numbers.

In fact, in modern computers, only arithmetical operations with finite numbers are realized. Numbers can be represented in computer systems in various ways using positional numeral systems with a finite radix $b$. We remind that *numeral* is a symbol or group of symbols that represents a *number*. The difference between numerals and numbers is the same as the difference between words and the things they refer to. A *number* is a concept that a *numeral* expresses. The same number can be represented by different numerals. For example, the symbols '3', 'three', and 'III' are different numerals, but they all represent the same number.

Usually, when mathematicians deal with infinite objects (sets or processes) it is supposed that human beings are able to execute certain operations infinitely many times. For example, in a fixed numeral system it is possible to write down a numeral with *any* number of digits. However, this supposition is an abstraction (courageously declared by constructivists e.g, in [7]) because we live in a finite world and all human beings and/or computers finish operations they have started.

The point of view proposed in this paper does not use this abstraction and, therefore, is closer to the world of practical calculus than the traditional approaches. On the one hand, we assume existence of infinite sets and processes. On the other hand, we accept that any of the existing numeral systems allows one to write down only a finite number of numerals and to execute a finite number of operations. Thus, the problem we deal with can be

formulated as follows: How to describe infinite sets and infinite processes by a finite number of symbols and how to execute calculations with them?

The second important point in the paper is linked to the latter part of this question. The goal of the paper is to construct a new numeral system that would allow us to introduce and to treat infinite and infinitesimal numbers in the same manner as we are used to do with finite ones, i.e., by applying the philosophical principle of Ancient Greeks '*the part is less than the whole*' which, in our opinion, very well reflects the world around us but is not incorporated in traditional infinity theories.

Of course, due to this more restrictive and applied statement, such concepts as bijection, numerable and continuum sets, cardinal and ordinal numbers cannot be used in this paper. However, the approach proposed here does not contradict Cantor. In contrast, it evolves his deep ideas regarding existence of different infinite numbers.

Let us start our consideration by studying situations arising in practice when it is necessary to operate with extremely large quantities (see [12] for a detailed discussion). Imagine that we are in a granary and the owner asks us to count how much grain he has inside it. There are a few possibilities of finding an answer to this question. The first one is to count the grain seed by seed. Of course, nobody can do this because the number of seeds is enormous.

To overcome this difficulty, people take sacks, fill them in with seeds, and count the number of sacks. It is important that nobody counts the number of seeds in a sack. At the end of the counting procedure, we shall have a number of sacks completely filled and some remaining seeds that are not sufficient to complete the next sack. At this moment it is possible to return to the seeds and to count the number of remaining seeds that have not been put in sacks (or a number of seeds that it is necessary to add to obtain the last completely full sack).

If the granary is huge and it becomes difficult to count the sacks, then trucks or even big train waggons are used. Of course, we suppose that all sacks contain the same number of seeds, all trucks – the same number of sacks, and all waggons – the same number of trucks. At the end of the counting we obtain a result in the following form: the granary contains 16 waggons, 19 trucks, 12 sacks, and 4 seeds of grain. Note, that if we add, for example, one seed to the granary, we can count it and see that the granary has more grain. If we take out one waggon, we again be able to say how much grain has been subtracted.

Thus, in our example it is necessary to count large quantities. They are finite but it is impossible to count them directly using elementary units of measure, $u_0$, i.e., seeds, because the quantities expressed in these units would be too large. Therefore, people are forced to behave as if the quantities were infinite.

To solve the problem of 'infinite' quantities, new units of measure, $u_1, u_2,$ and $u_3,$ are introduced (units $u_1$ – sacks, $u_2$ – trucks, and $u_3$ – waggons). The new units have the following important peculiarity: it is not known how many units $u_i$ there are in the unit $u_{i+1}$ (we do not count how many seeds are in a sack, we just *complete* the sack). Every unit $u_{i+1}$ is filled in completely by the units $u_i$. Thus, we know that all the units $u_{i+1}$ contain a certain number $K_i$ of units $u_i$ but this number, $K_i$, is unknown. Naturally, it is supposed that $K_i$ is the same for all instances of the units. Thus, numbers that it was impossible to express using only initial units of measure are perfectly expressible if new units are introduced.

This key idea of counting by introduction of new units of measure will be used in the paper to deal with infinite quantities. In Section 2, we introduce a new positional system with infinite radix allowing us to write down not only finite but infinite and infinitesimal numbers too. In Section 3, we describe arithmetical operations for all of them. Applications dealing with infinite sets, divergent series, and limits viewed from the positions of the new approach can be found in [12] and are not discussed in this paper. After all, Section 4 concludes the paper.

We conclude this Introduction by emphasizing that the goal of the paper is not to construct a complete theory of infinity and to discuss such concepts as, for example, 'set of all sets'. In contrast, the problem of infinity is considered from positions of applied mathematics and theory and practice of computations – fields being among the main scientific interests (see, for example, [12, 13]) of the author. A new viewpoint on infinity and the corresponding mathematical and computer science tools are introduced in the paper in order to give possibilities to solve applied problems.

## 2 Infinite and infinitesimal numbers

Different numeral systems have been developed by humanity to describe finite numbers. More powerful numeral systems allow one to write down more numerals and, therefore, to express more numbers. However, in all existing numeral systems allowing us to execute calculations numerals corresponding only to finite numbers are used. Thus, in order to have a possibility to write down infinite and infinitesimal numbers by a finite number of symbols, we need at least one new numeral expressing an infinite (or an infinitesimal) number. Then, it is necessary to propose a new numeral system fixing rules for writing down infinite and infinitesimal numerals and to describe arithmetical operations with them.

Note that introduction of a new numeral for expressing infinite and infinitesimal numbers is similar to introduction of the concept of zero and the numeral '0' that in the past have allowed people to develop positional systems being more powerful than numeral systems existing before.

In positional numeral systems fractional numbers are expressed by the record

$$(a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-(q-1)} a_{-q})_b \tag{1}$$

where numerals $a_i, -q \le i \le n$, are called *digits*, belong to the alphabet $\{0,1,\dots,b-1\}$, and the dot is used to separate the fractional part from the integer one. Thus, the numeral (1) is equal to the sum

$$a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + \dots + a_{-(q-1)} b^{-(q-1)} + a_{-q} b^{-q}. \tag{2}$$

In modern computers, the radix $b = 2$ with the alphabet $\{0,1\}$ is mainly used to represent numbers.

Record (1) uses numerals consisting of one symbol each, i.e., digits $a_i \in \{0,1,\dots,b-1\}$, to express how many finite units of the type $b^i$ belong to the number (2). Quantities of finite units $b^i$ are counted separately for each exponent $i$ and all symbols in the alphabet $\{0,1,\dots,b-1\}$ express finite numbers.

A new positional numeral system with infinite radix described in this section evolves the idea of separate count of units with different exponents used in traditional positional systems to the case of infinite and infinitesimal numbers. The infinite radix of the new system is introduced as the number of elements of the set $\mathbb{N}$ of natural numbers expressed by the numeral ① called *grossone*. This mathematical object is introduced by describing its properties postulated by the *Infinite Unit Axiom* consisting of three parts: Infinity, Identity, and Divisibility (we introduce them soon). This axiom is added to axioms for real numbers similarly to addition of the axiom determining zero to axioms of natural numbers when integer numbers are introduced. This means that it is postulated that associative and commutative properties of multiplication and addition, distributive property of multiplication over addition, existence of inverse elements with respect to addition and multiplication hold for grossone as for finite numbers.

Note that usage of a numeral indicating totality of the elements we deal with is not new in mathematics. It is sufficient to remind the theory of probability where events can be defined in two ways. First, as union of elementary events; second, as a sample space, $\Omega$, of all possible elementary events from where some elementary events have been excluded. Naturally, the second way to define events becomes particularly useful when the sample space consists of infinitely many elementary events.

*Infinity.* For any finite natural number $n$ it follows $n < ①$.

*Identity.* The following relations link ① to identity elements 0 and 1

$$0 \cdot ① = ① \cdot 0 = 0, \quad ① - ① = 0, \quad \frac{①}{①} = 1, \quad ①^0 = 1, \quad 1^{①} = 1. \tag{3}$$

*Divisibility.* For any finite natural number $n$ sets $\mathbb{N}_{k,n}, 1 \le k \le n$, being the $n$th parts of the set, $\mathbb{N}$, of natural numbers have the same number of elements indicated by the numeral $\frac{①}{n}$ where

$$\mathbb{N}_{k,n} = \{k, k+n, k+2n, k+3n, \dots\}, \qquad 1 \le k \le n, \qquad \bigcup_{k=1}^{n} \mathbb{N}_{k,n} = \mathbb{N}. \tag{4}$$

For example for $n = 1,2,3$ we have

$$\begin{aligned}
① \rightarrow \quad & \mathbb{N} = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ \dots\} \\
\frac{①}{2} \nearrow \quad & \mathbb{N}_{1,2} = \{1,\ \ 3,\ \ 5,\ \ 7,\ \dots\} \\
\frac{①}{2} \searrow \quad & \mathbb{N}_{2,2} = \{\ \ 2,\ \ 4,\ \ 6,\ \ \dots\} \\
& \mathbb{N}_{1,3} = \{1,\ \ \ \ 4,\ \ \ \ 7,\ \dots\} \\
\frac{①}{3} \rightarrow \quad & \mathbb{N}_{2,3} = \{\ \ 2,\ \ \ \ 5,\ \ \ \ \dots\} \\
& \mathbb{N}_{3,3} = \{\ \ \ \ 3,\ \ \ \ 6,\ \ \ \dots\}
\end{aligned}$$

Before the introduction of the new positional system let us see some properties of grossone. Its role in infinite arithmetic is similar to the role of the number 1 in the finite one and it will serve us as the basis for construction of other infinite numbers. It is important to emphasize that to introduce $\frac{①}{n}$ we do not try to count elements $k, k+n, k+2n, k+3n, \dots$ In fact, we cannot do this since our possibilities to count are limited and, therefore,

we are not able to count for infinity. In contrast, we postulate following the mentioned above Ancient Greeks' principle *the part is less than the whole* (see [1, 9, 12] for detailed discussions on such a kind of approaches) that the number of elements of the *n*th part of the set, i.e., $\frac{①}{n}$, is *n* times less than the number of elements of the whole set, i.e., than ①. In terms of our granary example ① can be interpreted as the number of seeds in the sack. Then, if the sack contains ① seeds, its *n*th part contains $\frac{①}{n}$ seeds.

The numbers $\frac{①}{n}$ have been introduced as numbers of elements of sets $\mathbb{N}_{k,n}$ thus, they are integer. For example, due to the introduced axiom, the set

$$\mathbb{N}_{2,5} = \{2, 7, 12, \ldots\}$$

has $\frac{①}{5}$ elements and the set

$$\mathbb{N}_{3,10} = \{3, 13, 23, \ldots\}$$

has $\frac{①}{10}$ elements.

The number of elements of sets being union, intersection, difference, or product of other sets of the type $\mathbb{N}_{k,n}$ is defined in the same way as these operations are defined for finite sets. Thus, we can define the number of elements of sets being results of these operations with finite sets and infinite sets of the type $\mathbb{N}_{k,n}$. For instance, the number of elements of the set

$$\mathbb{N}_{2,5} \cup \mathbb{N}_{3,10} \cup \{2, 3, 4, 5\}$$

is $\frac{①}{5} + \frac{①}{10} + 2$ because

$$\mathbb{N}_{2,5} \cap \mathbb{N}_{3,10} = \varnothing, \quad 2 \in \mathbb{N}_{2,5}, \quad 3 \in \mathbb{N}_{3,10}.$$

It is worthwhile to notice that, as it is for finite sets, infinite sets constructed using finite sets and infinite sets of the type $\mathbb{N}_{k,n}$ have the same number of elements independently of objects outside the sets. A general rule for determining the number of elements of infinite sets having a more complex structure can be also given but it is not discussed in this paper (see [12]).

Introduction of the numeral ① allows us to write down the set, $\mathbb{N}$, of natural numbers in the form

$$\mathbb{N} = \{1, 2, 3, \quad \ldots \quad ① - 2, \ ① - 1, \ ①\}. \tag{5}$$

It is worthwhile to notice that the set (5) is the same set of natural numbers we are used to deal with. We have introduced grossone as the quantity of natural numbers. Thus, it is the biggest natural number and numbers

$$\ldots \ ① - 3, \ ① - 2, \ ① - 1 \tag{6}$$

less than grossone are natural numbers as the numbers $1, 2, 3, \ldots$ The difficulty to accept existence of infinite natural numbers is in the fact that traditional numeral systems did not allow us to see these numbers. Similarly, primitive tribes working with unary numeral system were able to see only numbers 1, 2, and 3 because they operated only with numerals *I*, *II*, *III* and did not suspect about existence of other natural numbers. For them, all quantities bigger than *III* were just 'many' and such operations as $II + III$ and $I + III$ give the same result, i. e., 'many'. Note that this happens not because $II + III = I + III$ but due to weakness of this primitive numeral system. This weakness leads also to such results as 'many'+1 ='many' and 'many'+2 ='many' which are very familiar to us in the context of views on infinity used in the traditional calculus: $\infty + 1 = \infty, \infty + 2 = \infty$.

As an example let us consider a numeral system $\mathcal{S}$ able to express only numbers 1 and 2 by the numerals '1' and '2' (this system is even simpler than that of primitive tribes which was able to express three natural numbers). If we add to this system the new numeral ① it becomes possible to express the following numbers

$$\underbrace{1, 2}_{finite}, \quad \ldots \quad \underbrace{\frac{①}{2} - 2, \frac{①}{2} - 1, \frac{①}{2}, \frac{①}{2} + 1, \frac{①}{2} + 2}_{infinite}, \quad \ldots \quad \underbrace{① - 2, ① - 1, ①}_{infinite}.$$

In this record the first two numbers are finite, the remaining eight are infinite, and dots show the natural numbers that are not expressible in this numeral system. This numeral system does not allow us to execute such operations as $2 + 2$ or $2 + \frac{①}{2} + 2$ because their results cannot be expressed in this system but, of course, we do not write that results of these operations are equal, we just say that the results are not expressible in this numeral system and it is necessary to take another, more powerful one.

The introduction of grossone allows us to obtain the following interesting result: the set $\mathbb{N}$ is not a monoid under addition. In fact, the operation $① + 1$ gives us as the result a number grater than ①. Thus, by definition of grossone, $① + 1$ does not belong to $\mathbb{N}$ and, therefore, $\mathbb{N}$ is not closed under addition and is not a monoid.

This result also means that adding the Infinite Unit Axiom to the axioms of natural numbers defines the set of *extended natural numbers* indicated as $\widehat{\mathbb{N}}$ and including $\mathbb{N}$ as a proper subset

$$\widehat{\mathbb{N}} = \{1, 2, \ldots, ① - 1, ①, ① + 1, \ldots, ①^2 - 1, ①^2, ①^2 + 1, \ldots\}.$$

Again, extended natural numbers grater than grossone can also be interpreted in the terms of sets of numbers. For example, $① + 3$ as the number of elements of the set $\mathbb{N} \cup \{a, b, c\}$ where numbers $a, b, c \notin \mathbb{N}$ and $①^2$ as the number of elements of the set $\mathbb{N} \times \mathbb{N}$. In terms of our granary example $① + 3$ can be interpreted as a sack plus three seeds and $①^2$ as a truck.

We have already started to write down simple infinite numbers and to execute arithmetical operations with them without concentrating our attention upon this question. Let us consider it systematically.

To express infinite and infinitesimal numbers we shall use records that are similar to (1) and (2) but have some peculiarities. In order to construct a number $C$ in the new numeral positional system with base ① we subdivide $C$ into groups corresponding to powers of ①:

$$C = c_{p_m}①^{p_m} + \ldots + c_{p_1}①^{p_1} + c_{p_0}①^{p_0} + c_{p_{-1}}①^{p_{-1}} + \ldots + c_{p_{-k}}①^{p_{-k}}. \tag{7}$$

Then, the record

$$C = c_{p_m}①^{p_m} \ldots c_{p_1}①^{p_1} c_{p_0}①^{p_0} c_{p_{-1}}①^{p_{-1}} \ldots c_{p_{-k}}①^{p_{-k}}. \tag{8}$$

represents the number $C$, symbols $c_i$ are called *grossdigits*, symbols $p_i$ are called *grosspowers*. The numbers $p_i$ are such that $p_i > 0, p_0 = 0, p_{-i} < 0$ and

$$p_m > p_{m-1} > \ldots p_2 > p_1 > p_{-1} > p_{-2} > \ldots p_{-(k-1)} > p_{-k}.$$

In the traditional record (1) there exists a convention that a digit $a_i$ shows how many powers $b^i$ are present in the number and the radix $b$ is not written explicitly. In the record (8) we write $①^{p_i}$ explicitly because in the new numeral positional system the number $i$ in general is not equal to the grosspower $p_i$. This gives possibility to write, for example, such numbers as $7①^{244.5}3①^{-32}$ where $p_1 = 244.5, p_{-1} = -32$.

Finite numbers in this new numeral system are represented by numerals having only one grosspower equal to zero. In fact, if we have a number $C$ such that $m = k = 0$ in representation (8), then due to (3) we have $C = c_0①^0 = c_0$. Thus, the number $C$ in this case does not contain infinite and infinitesimal units and is equal to the grossdigit $c_0$ which being a conventional finite number can be expressed in the form (1), (2) by any positional system with finite base $b$ (or by another numeral system). It is important to emphasize that the grossdigit $c_0$ can be integer or fractional and can be expressed by a few symbols in contrast to the traditional record (1) where each digit is integer and is represented by one symbol from the alphabet $\{0, 1, 2, \ldots, b-1\}$. Thus, the grossdigit $c_0$ shows how many finite units and/or parts of the finite unit, $1 = ①^0$, there are in the number $C$. Grossdigits can be written in positional systems, in the form $\frac{p}{q}$ where $p$ and $q$ are integer numbers, or in any other finite numeral system.

Analogously, in the general case, all grossdigits $c_i, -k \leq i \leq m$, can be integer or fractional and expressed by many symbols. For example, the number $\frac{7}{3}①^4\,\frac{84}{19}①^{-3.1}$ has grossdigits $c_4 = \frac{7}{3}$ and $c_{-3.1} = \frac{84}{19}$. All grossdigits can also be negative; they show how many corresponding units should be added or subtracted in order to form the number $C$.

Infinite numbers are written in this numeral system as numerals having grosspowers grater than zero, for example $7①^{244.5}3①^{-32}$ and $-2①^{74}3①^{0}37①^{-2}11①^{-15}$ are infinite numbers. In the following example the left-hand expression presents the way to write down infinite numbers and the right-hand shows how the value of the number is calculated:

$$15①^{14}17.2045①^3(-52.1)①^{-6} = 15①^{14} + 17.2045①^3 - 52.1①^{-6}.$$

If a grossdigit $c_{p_i}$ is equal to 1 then we write $①^{p_i}$ instead of $1①^{p_i}$. Analogously, if power $①^0$ is the lowest in a number then we often use simply the corresponding grossdigit $c_0$ without $①^0$, for instance, we write $23①^{14}5$ instead of $23①^{14}5①^0$ or 3 instead of $3①^0$.

Numerals having only negative grosspowers represent infinitesimal numbers. The simplest number from this group is $①^{-1} = \frac{1}{①}$ being the inverse element with respect to multiplication for ①:

$$\frac{1}{①} \cdot ① = ① \cdot \frac{1}{①} = 1. \tag{9}$$

Note that all infinitesimals are not equal to zero. Particularly, $\frac{1}{①} > 0$ because $1 > 0$ and $① > 0$. It has a clear interpretation in our granary example. Namely, if we have a sack and it contains ① seeds then one sack divided by ① is equal to one seed. Vice versa, one seed, i.e., $\frac{1}{①}$, multiplied by the number of seeds in the sack, ①, gives one sack of seeds.

Inverse elements of more complex numbers including grosspowers of ① are defined by a complete analogy. The following two numbers are examples of infinitesimals $3①^{-32}$, $37①^{-2}(-11)①^{-15}$.

## 3 Arithmetical operations with infinite, infinitesimal, and finite numbers

Let us now introduce arithmetical operations for infinite, infinitesimal, and finite numbers. The operation of *addition* of two given infinite numbers $A$ and $B$ (the operation of subtraction is a direct consequence of that of addition and is thus omitted) returns as the result an infinite number $C$

$$A=\sum_{i=1}^{K} a_{k_i}①^{k_i}, \qquad B=\sum_{j=1}^{M} b_{m_j}①^{m_j}, \qquad C=\sum_{i=1}^{L} c_{l_i}①^{l_i}, \tag{10}$$

where $C$ is constructed by including in it all items $a_{k_i}①^{k_i}$ from $A$ such that $k_i \neq m_j, 1 \leq j \leq M$, and all items $b_{m_j}①^{m_j}$ from $B$ such that $m_j \neq k_i, 1 \leq i \leq K$. If in $A$ and $B$ there are items such that $k_i = m_j$ for some $i$ and $j$ then this grosspower $k_i$ is included in $C$ with the grossdigit $b_{k_i}+a_{k_i}$, i.e., as $(b_{k_i}+a_{k_i})①^{k_i}$. It can be seen from this definition that the introduced operation enjoys the usual properties of commutativity and associativity due to definition of grossdigits and the fact that addition for each grosspower of ① is executed separately.

Let us illustrate the rules by an example (in order to simplify the presentation in all the following examples the radix $b=10$ is used for writing down grossdigits). We consider two infinite numbers $A$ and $B$ where

$$A=16.5①^{44.2}(-12)①^{12}17①^{0}1.17①^{-3},$$

$$B=23①^{14}6.23①^{3}10.1①^{0}(-1.17)①^{-3}11①^{-43}.$$

Their sum $C$ is calculated as follows

$$C=A+B=16.5①^{44.2}+(-12)①^{12}+17①^{0}+1.17①^{-3}+$$

$$23①^{14}+6.23①^{3}+10.1①^{0}-1.17①^{-3}+11①^{-43}=$$

$$16.5①^{44.2}+23①^{14}-12①^{12}+6.23①^{3}+$$

$$(17+10.1)①^{0}+(1.17-1.17)①^{-3}+11①^{-43}=$$

$$16.5①^{44.2}+23①^{14}-12①^{12}+6.23①^{3}+27.1①^{0}+11①^{-43}=$$

$$16.5①^{44.2}23①^{14}(-12)①^{12}6.23①^{3}27.1①^{0}11①^{-43}.$$

The operation of *multiplication* of two given infinite numbers $A$ and $B$ from (10) returns as the result the infinite number $C$ constructed as follows.

$$C=\sum_{j=1}^{M} C_j, \qquad C_j = b_{m_j}①^{m_j}\cdot A=\sum_{i=1}^{K} a_{k_i}b_{m_j}①^{k_i+m_j}, \qquad 1\leq j\leq M. \tag{11}$$

Similarly to addition, the introduced multiplication is commutative and associative. It is easy to show that the distributive property is also valid for these operations.

Let us illustrate this operation by the following example. We consider two infinite numbers

$$A=①^{18}(-5)①^{2}(-3)①^{1}0.2, \qquad B=①^{2}(-1)①^{1}7①^{-3}$$

and calculate the product $C=B\cdot A$. The first partial product $C_1$ is equal to

$$C_1=7①^{-3}\cdot A=7①^{-3}(①^{18}-5①^{2}-3①^{1}+0.2)=$$

$$7①^{15}-35①^{-1}-21①^{-2}+1.4①^{-3}=7①^{15}(-35)①^{-1}(-21)①^{-2}1.4①^{-3}.$$

The other two partial products, $C_2$ and $C_3$, are computed analogously:

$$C_2=-①^{1}\cdot A=-①^{1}(①^{18}-5①^{2}-3①^{1}+0.2)=-①^{19}5①^{3}3①^{2}(-0.2)①^{1},$$

$$C_3=①^{2}\cdot A=①^{2}(①^{18}-5①^{2}-3①^{1}+0.2)=①^{20}(-5)①^{4}(-3)①^{3}0.2①^{2}.$$

Finally, by taking into account that grosspowers $①^3$ and $①^2$ belong to both $C_2$ and $C_3$ and, therefore, it is necessary to sum up the corresponding grossdigits, the product $C$ is equal (due to its length, the number $C$ is written in two lines) to

$$C=C_1+C_2+C_3=①^{20}(-1)①^{19}7①^{15}(-5)①^{4}2①^{3}3.2①^{2}$$
$$(-0.2)①^{1}(-35)①^{-1}(-21)①^{-2}1.4①^{-3}.$$

In the operation of *division* of a given infinite number $C$ by an infinite number $B$ we obtain an infinite number $A$ and a reminder $R$ that can be also equal to zero, i.e., $C = A \cdot B + R$.

The number $A$ is constructed as follows. The numbers $B$ and $C$ are represented in the form (10). The first grossdigit $a_{k_K}$ and the corresponding maximal exponent $k_K$ are established from the equalities

$$a_{k_K} = c_{l_L} / b_{m_M}, \quad k_K = l_L - m_M. \tag{12}$$

Then the first partial reminder $R_1$ is calculated as

$$R_1 = C - a_{k_K} ①^{k_K} \cdot B. \tag{13}$$

If $R_1 \neq 0$ then the number $C$ is substituted by $R_1$ and the process is repeated by a complete analogy. The grossdigit $a_{k_{K-i}}$, the corresponding grosspower $k_{K-i}$ and the partial reminder $R_{i+1}$ are computed by formulae (14) and (15) obtained from (12) and (13) as follows: $l_L$ and $c_{l_L}$ are substituted by the highest grosspower $n_i$ and the corresponding grossdigit $r_{n_i}$ of the partial reminder $R_i$ that in its turn substitutes $C$:

$$a_{k_{K-i}} = r_{n_i} / b_{m_M}, \quad k_{K-i} = n_i - m_M. \tag{14}$$

$$R_{i+1} = R_i - a_{k_{K-i}} ①^{k_{K-i}} \cdot B, \quad i \geq 1. \tag{15}$$

The process stops when a partial reminder equal to zero is found (this means that the final reminder $R = 0$) or when a required accuracy of the result is reached.

The operation of division will be illustrated by two examples. In the first example we divide the number $C = -10①^3 16①^0 42①^{-3}$ by the number $B = 5①^3 7$. For these numbers we have

$$l_L = 3, \;\; m_M = 3, \;\; c_{l_L} = -10, \;\; b_{m_M} = 5.$$

It follows immediately from (12) that $a_{k_K} ①^{k_K} = -2①^0$. The first partial reminder $R_1$ is calculated as

$$R_1 = -10①^3 16①^0 42①^{-3} - (-2①^0) \cdot 5①^3 7 =$$

$$-10①^3 16①^0 42①^{-3} + 10①^3 14①^0 = 30①^0 42①^{-3}.$$

By a complete analogy we should construct $a_{k_{K-1}} ①^{k_{K-1}}$ by rewriting (12) for $R_1$. By doing so we obtain equalities

$$30 = a_{k_{K-1}} \cdot 5, \quad 0 = k_{K-1} + 3$$

and, as the result, $a_{k_{K-1}} ①^{k_{K-1}} = 6①^{-3}$. The second partial reminder is

$$R_2 = R_1 - 6①^{-3} \cdot 5①^3 7 = 30①^0 42①^{-3} - 30①^0 42①^{-3} = 0.$$

Thus, we can conclude that the reminder $R = R_2 = 0$ and the final result of division is $A = -2①^0 6①^{-3}$.

Let us now substitute the grossdigit 42 by 40 in $C$ and divide this new number $\widetilde{C} = -10①^3 16①^0 40①^{-3}$ by the same number $B = 5①^3 7$. This operation gives us the same result $\widetilde{A}_2 = A = -2①^0 6①^{-3}$ (where subscript 2 indicates that two partial reminders have been obtained) but with the reminder $\widetilde{R} = \widetilde{R}_2 = -2①^{-3}$. Thus, we obtain $\widetilde{C} = B \cdot \widetilde{A}_2 + \widetilde{R}_2$. If we want to continue the procedure of division, we obtain $\widetilde{A}_3 = -2①^0 6①^{-3} (-0.4)①^{-6}$ with the reminder $\widetilde{R}_3 = 0.28①^{-6}$. Naturally, it follows $\widetilde{C} = B \cdot \widetilde{A}_3 + \widetilde{R}_3$. The process continues until a partial reminder $\widetilde{R}_i = 0$ is found or when a required accuracy of the result will be reached.

In all the examples above we have used grosspowers being finite numbers. However, all the arithmetical operations work by a complete analogy also for grosspowers being themselves numbers of the type (8). For example, if

$$X = 16.5①^{44.2①^{1.17①^{-3}}} (-12)①^{12①} 1.17①^{-3},$$

$$Y = 23①^{44.2①^{1.17①^{-3}}} (-1.17)①^{-3} 11①^{4①^{-23}},$$

then their sum $Z$ is calculated as follows

$$Z = X + Y = 39.5①^{44.2①^{1.17①^{-3}}} (-12)①^{12①} 11①^{4①^{-23}}.$$

## 4 Conclusion

In this paper, a new positional numeral system with infinite radix has been described. This system allows us to express by a finite number of symbols not only finite numbers but infinite and infinitesimals too. All of them can be viewed as particular cases of a general framework used to express numbers.

It has been shown in [11] that the new approach allows us to construct a new type of computer – Infinity Computer – able to operate with finite, infinite, and infinitesimals quantities. Numerous examples dealing with infinite sets and processes, divergent series, limits, and measure theory viewed from the positions of the new approach can be found in [12]. We conclude this paper just by one example showing the potential of the new approach. We consider two infinite series $S_1 = 1+1+1+\ldots$ and $S_2 = 3+3+3+\ldots$ The traditional analysis gives us a very poor answer that both of them diverge to infinity. Such operations as $S_1 - S_2$ or $\frac{S_1}{S_2}$ are not defined.

In our terminology divergent series do not exist. Now, when we are able to express not only different finite numbers but also different infinite numbers, the records $S_1$ and $S_2$ are not well defined. It is necessary to indicate explicitly the number of items in the sum and it is not important is it finite or infinite. To calculate the sum it is necessary that the number of items and the result are expressible in the numeral system used for calculations. It is important to notice that even though a sequence cannot have more than ① elements the number of items in a series can be greater than grossone because the process of summing up is not necessary should be executed by a sequential adding items.

Suppose that the series $S_1$ has $k$ items and $S_2$ has $n$ items:

$$S_1(k) = \underbrace{1+1+1+\ldots+1}_{k}, \qquad S_2(n) = \underbrace{3+3+3+\ldots+3}_{n}.$$

Then $S_1(k) = k$ and $S_2(n) = 3n$ for any values of $k$ and $n$ – finite or infinite. By giving numerical values to $k$ and $n$ we obtain numerical values for the sums. If, for instance, $k = n = 5①$ then we obtain $S_1(5①) = 5①$, $S_2(5①) = 15①$ and

$$S_2(5①) - S_1(5①) = 10① > 0.$$

If $k = 5①$ and $n = ①$ we obtain $S_1(5①) = 5①$, $S_2(①) = 3①$ and it follows

$$S_2(①) - S_1(5①) = -2① < 0.$$

If $k = 3①$ and $n = ①$ we obtain $S_1(3①) = 3①$, $S_2(①) = 3①$ and it follows

$$S_2(①) - S_1(3①) = 0.$$

Analogously, the expression $\frac{S_1(k)}{S_2(n)}$ can be calculated.

## References


1. V. Benci and M. Di Nasso, *Numerosities of labeled sets: a new way of counting*, Advances in Mathematics **173** (2003), 50–67.
2. G. Cantor, *Contributions to the founding of the theory of transfinite numbers*, Dover Publications, New York, 1955.
3. P. J. Cohen, *Set theory and the Continuum Hypothesis*, Benjamin, New York, 1966.
4. J. H. Conway and R. K. Guy, *The book of numbers*, Springer-Verlag, New York, 1996.
5. K. Gödel, *The consistency of the Continuum-Hypothesis*, Princeton University Press, Princeton, 1940.
6. D. Hilbert, *Mathematical problems: Lecture delivered before the International Congress of Mathematicians at Paris in 1900*, Bulletin of the American Mathematical Society **8** (1902), 437–479.
7. A. A. Markov Jr. and N. M. Nagorny, *Theory of algorithms*, second ed., FAZIS, Moscow, 1996.
8. P. A. Loeb and M. P. H. Wolff, *Nonstandard analysis for the working mathematician*, Kluwer Academic Publishers, Dordrecht, 2000.
9. J. P. Mayberry, *The foundations of mathematics in the theory of sets*, Cambridge University Press, Cambridge, 2001.
10. A. Robinson, *Non-standard analysis*, Princeton University Press, Princeton, 1996.
11. Ya. D. Sergeyev, *Computer system for storing infinite, infinitesimal, and finite quantities and executing arithmetical operations with them*, patent application, 08.03.04.
12. _____, *Arithmetic of infinity*, Edizioni Orizzonti Meridionali, CS, 2003.
13. R. G. Strongin and Ya. D. Sergeyev, *Global optimization and non-convex constraints: Sequential and parallel algorithms*, Kluwer Academic Publishers, Dordrecht, 2000.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.9-18
ISBN 83-922646-0-6


# Towards Wireless Sensor Networks with Enhanced Vision Capabilities

Andrzej Śluzek[1,2], Palaniappan Annamalai[1], Md. Saiful Islam[1], Pawel Czapski[1]
[1]Nanyang Technological University, Singapore
[2]SWPS, Warszawa, Poland

**Abstract.** Wireless sensor networks are expected to become an important tool for various security, surveillance and/or monitoring applications. The paper discusses selected practical aspects of development of such networks. First, design and implementation of an exemplary wireless sensor network for intrusion detection and classification is briefly presented. The network consists of two levels of nodes. At the first level, relatively simple microcontroller-based nodes with basic sensing devices and wireless transmission capabilities are used. These nodes are used as preliminary detectors of prospective intrusions. The second-level sensor node is built around a high performance FPGA controlling an array of cameras. The second-level nodes can be dynamically reconfigured to perform various types of visual data processing and analysis algorithms used to confirm the presence of intruders in the scene and to approximately classify the intrusion, if any. The paper briefly presents algorithms and overviews hardware of the network. In the last part of the paper, prospective directions for wireless sensor networks are analyzed and certain recommendations are proposed.

## 1. Introduction

Adequate sensing capabilities are the key issue in a system working autonomously or semi-autonomously in unstructured, unfamiliar environments. Wireless sensor networks are one of the emerging technologies for providing such systems with information about current conditions in a large fragment of the environment. In such networks, a huge amount of sensed data should be acquired and processed in real time. Thus, adequate hardware resources capable of real-time data processing and intelligent data selection mechanisms are needed in order to prevent informational and communicational saturation of the network [1].

Wireless sensor networks are being used for various condition-based monitoring tasks such as habitat monitoring, environmental monitoring [2], machineries and aerospace monitoring and for security and surveillance [3, 4]. Such applications need tens to several hundreds of nodes. Each node can be equipped with various sensors (e. g. proximity, temperature, acoustic, vibration) to monitor conditions of the environment and possibly to fulfill other requirements of the task. Additionally, image sensors are usually present in security and surveillance applications where more detailed analysis of the environment is expected.

The most important challenge in such systems is to manage the huge information flow in the network nodes. The systems with various non-visual and visual sensors acquire and should process in real-time large amounts of data. Transmitting raw data from every node to other nodes and/or to the higher level occupies the bandwidth for a long period of time. This in turn reduces the reporting efficiency of the system by delaying or loosing messages about important event happening in the environment. Moreover, most of the raw data transmitted from the sensor nodes is unimportant, so that the energy (which is a precious commodity in autonomous systems) is wasted for communication and flooding the destination node with useless data.

The most straightforward solution is to provide the sensor nodes with a certain level of autonomous intelligence so that the captured data can be locally processed and analyzed, while only observation of significant importance or unknown situations would be reported to the higher level.

Initially, wireless sensor network were defined (mostly because of the envisaged military applications) as large-scale, wireless, ad-hoc, multi-hop, unpartitioned networks of homogenous, tiny, and immobile sensor nodes. It has been shown by many non-military applications, however, that the above assumptions are inaccurate [14]. Wireless sensor networks can be heterogeneous, possibly mobile, and with different network topologies. Generally, their nodes have communicational, computational and sensing capabilities, though in diversified proportions [15].

We have implemented the concept of a heterogeneous sensor network in an experimental platform eventually intended for various military and civilian applications where a visual surveillance of large areas is required. Visual surveillance in considered the most effective method of monitoring complex environments, but systems that could perform such a task fully autonomously and reliably are still very difficult to build. Visual assessment of a situation by a human is still considered the ultimate factor in taking important decisions. In complex scenarios, however, the amount of data transmitted across the network would make the human inspection and/or assessment of the situation very difficult. Thus, we have proposed a realistic approach, where a human operator can be supported by vision capabilities of the network that can automatically handle typical situations. The human

intervention is needed only in special situation (or situations involving decisions which are reserved for humans).

The local computational intelligence of the selected nodes has been achieved by incorporating FPGA (field programmable gate array) devices into selected nodes. These FPGA based nodes can process the data received from other nodes and process the images in order to analyze the situation. The results are wirelessly transmitted to the higher level in the decision chain only if there is any intrusion or another unusual event so that the human operator can confirm (or reject) the system's decision that a potential danger has been detected.

This paper presents design methodology (for both hardware architecture and the embedded algorithms) for the current system and summarizes proposals for the future wireless sensor networks with enhanced vision capabilities. In Section 2, we overview the design of the network nodes. Section 3 presents image processing algorithms implemented in the FPGA nodes for intrusion detection and classification. Additional implementation issues are discussed in Section 4. The final Section 5 summarizes the results and highlights selected problems important for the future development of wireless sensor networks with enhanced vision capabilities. Unfortunately, some implementation details are not disclosed in this paper as the developed system is a feasibility study for a commercial product.

## 2. Architecture of the network

In typical wireless sensor applications, the network nodes have to process all the data from a variety of sensors and at the same time have to efficiently manage the power to achieve a reasonably long operational lifetime. Various architectures for different applications have been proposed by researchers in the recent years (e. g. [5, 6, 7, 8]) but only a few papers discuss the power management in the wireless sensor networks in the context of the overall system design.

In the developed platform, two different types of nodes are used. The first level nodes are relatively simple with basic sensing devices (e. g. proximity, vibration, acoustic, magnetic sensors) and wireless transmission capabilities. They continuously monitor conditions in the protected zone and act as preliminary detectors of possible intrusions. These sensor nodes are built around a simple micro-controller so that they consume very low power but their computational power is also very low. The second level sensor nodes are built around a higher performance FPGA controlling an array of cameras. They perform more advanced data processing to confirm or reject the intrusion (before alerting a human operator). Each camera is activated after the corresponding first-level nodes acquire data that may indicate a presence of an intruder, and transmit the warning message in wireless network to the FPGA based second level nodes. The overview of the network structure is shown in Fig. 1.

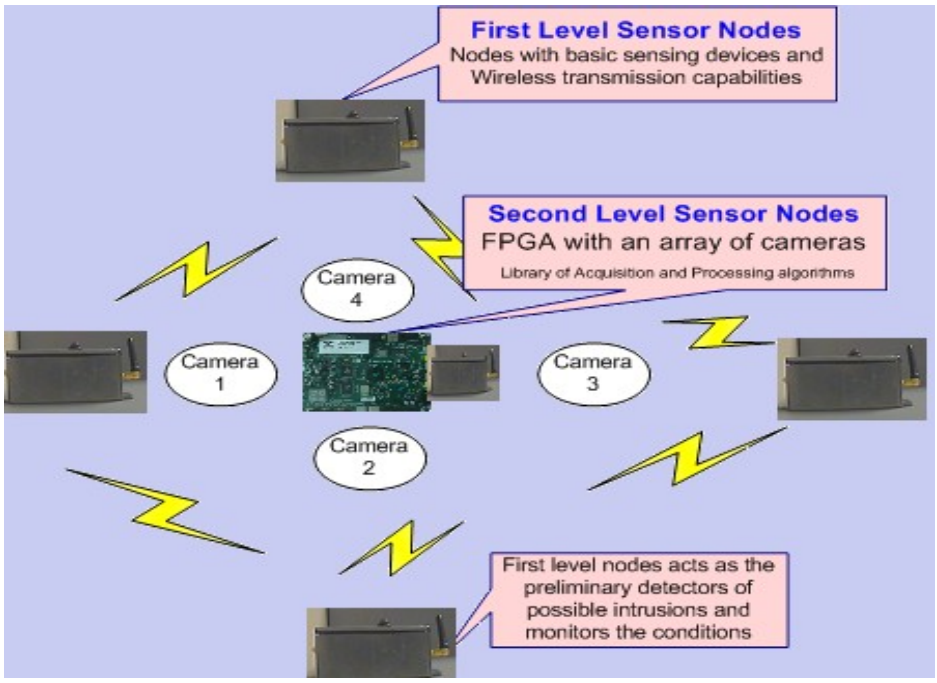



**Fig. 1. Basic structure of the developed sensor network.**

### 2.1 First Level Nodes

The first-level nodes (Fig. 2) incorporate a wireless communication chip, a low-cost microcontroller (performing sensor data acquisition, generating wirelessly-transmitted messages and interfacing the wireless chip) and basic non-vision sensors. Battery power supply is provided. Currently, only infrared proximity sensors and vibrations sensors are used, but additional options are planned (and the corresponding sensors are already available in the nodes).

**Fig. 2. The first-level node of the network.**

The first level nodes generally use very simple detection technique. For example, analog signals are just thresholded and only limited signal processing capabilities are available. They are assumed to be continuously active. For the first level nodes, a possible intrusion is defined as a particular coincidence of sensor readings. Different definitions of various intrusion types can be used, either for different applications or within the same application. Whenever a possible intrusion is sensed, the node wirelessly transmits a message containing the node identifier and (if several definitions of possible intrusions are simultaneously used) the type of sensed intrusion. To provide a higher level of reliability, the message may be repeated several times. A standard communication protocol for low-power wireless networks is used. The intended message recipients are the second level nodes (those close enough to receive the message) which would perform more detailed analysis of the event.

The first level nodes consume low power as they only transmit short messages (only if the sensors detect anything unusual) and perform very little computations. Low energy consumption is a crucial issue since a very long operational lifetime is expected [6]. No online reprogrammability/reconfigurability is envisaged and these first level nodes are assumed cheap expendables.

### 2.2 Second Level Nodes

The second-level nodes area built around an FPGA module that can control up to four cameras. Additionally, the nodes are equipped with the same wireless communication components as the first-level nodes. Typically, one second-level node is associated with several first-level nodes, but the network structure is not permanent. We envisage that eventually ad hoc self-organization mechanisms will be used during the network deployment (e. g. [16]).

In the second-level nodes, the power consumption by both FPGA and cameras is relatively high. Therefore, only the kernel of a second-level node would be permanently active, while the other parts (e. g. the cameras) are activated only when a warning message from the first level is received. Two option are possible: (1) a second-level node can be activated by any first-level node within the wireless range or (2) a second-level level node can be activated by selected first-level nodes (i. e. the warning messages containing identifiers of other nodes are ignored).

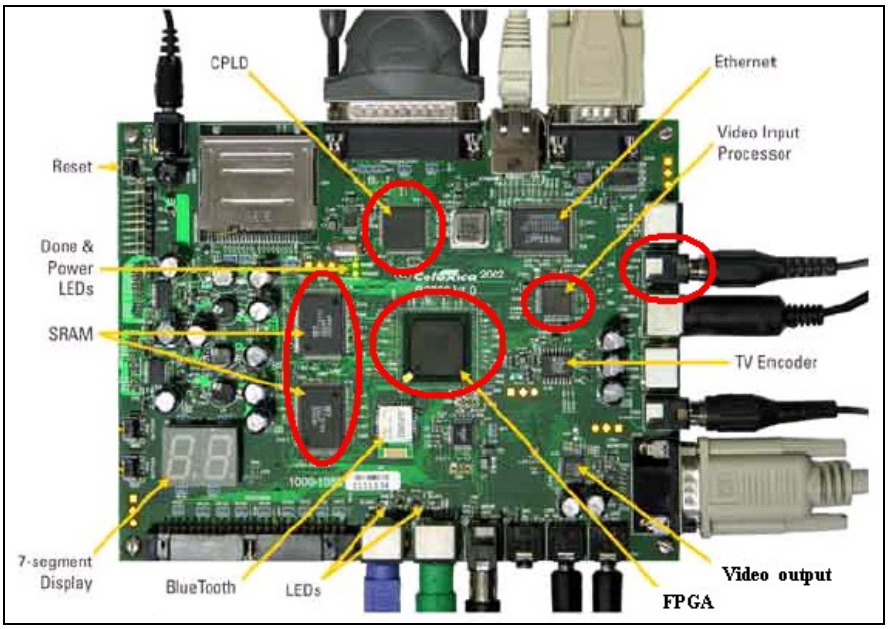



**Fig. 3. Components of the FPGA development board used in the second-level node.**

Upon activation, a second-level node camera captures a short sequence of images (typically two or three) that are subsequently processed using dedicated algorithms implemented in the node's FPGA. In general, the purpose of image processing is to extract the possible intruders from a captured image and to classify/identify them. After the task is completed, selected fragments of camera-captured images and/or other results produced by the algorithms may be wirelessly transmitted to the higher level in the decision chain (possibly including a human operator). Additionally, the second-level nodes can be periodically activated in order to update the background image (see Section 3).

In general, the peak energy consumption in a second-level node may be high, but the average power requirements could be low enough to provide long operational lifetime. In the prototype platform, power is supplied by a DC adaptor.

The prototype second-level nodes are built around a commercially available FPGA development board (see Fig. 3) with a powerful Virtex II chip (8,000,000 equivalent gates). However, only selected resources available in the board are used in the prototype (FPGA chip with external memory, CPLD module, CCD cameras and camera interfaces as shown in Fig. 3). Moreover, the overall usage of the FPGA capacity does not exceed 15% so that many other tasks can be simultaneously implemented within the same FPGA. Typically, these additional tasks are various visualization routines allowing testing and performance assessment of the node. In the final version, a much less powerful FPGA is envisaged. These nodes can be motionless or attached to a mobile platform.

## 3. Image Processing in Intrusion Detection and Classification

This section discusses the exemplary methodologies for FPGA-implemented image processing and intrusion detection and classification. Various systems used for surveillance, tracking and monitoring applications have been discussed in [3, 4, 7]. In our system, however, the camera-captured images are acquired, processed and analyzed by FPGA. The results are sent in a visual form to a human operator only if the system suspects something unusual is happening in the monitored area. Various image processing algorithms have been implemented in FPGA to perform different tasks. The library of developed algorithms can be used for adapting the network for changing conditions and/or requirements. Through online wireless reconfiguration of FPGA, the same hardware platform can adapted to various tasks without redeploying the network.

The fundamental task of the developed network is to detect visible changes in the monitored area, i. e. to identify non-stationary (moving) components within camera-captured images. Generally, there are three typical approaches to the extraction of moving targets from a sequence of images (or a video stream) as discussed in [3]. These are: (1) temporal differencing (two or three frame), (2) background subtraction and (3) optical flow. Temporal differencing is very adaptive to dynamic environments but does not perform well in extracting all

relevant feature pixels. Back ground subtraction is very sensitive to dynamic environment changes but provides complete feature data. Optical flow computation methods are considered too complex for real-time applications unless a specialized hardware is available. For our system, background subtraction has been selected. Although such an operation can be (hypothetically) implemented as a simple image comparison, a more sophisticated method is needed for more realistic applications. In general, the scene background may be affected by both illumination variations and minor configuration changes (e. g. swaying trees, grass, vibrations of the network nodes, etc.). The first problem can be solved by occasional background updates (see the later part of this section) but for the second one, a more sophisticated approach has been proposed. In the developed system, we apply a combination of subtractions in RGB space (with a threshold defining the acceptable differences) followed by a sequence of selected morphological operations. The purpose of the morphological operations is to filter out differences caused by minor configuration changes.

### 3.1 Static Intrusion Detection

Silhouettes of intruders obtained by background subtraction can be represented in three different forms: (1) a binary blob of the intruder's shape, (2) a full image within the extracted silhouette or (3) a rectangular outline of the intruder. Examples of the original scene, the scene with the intruder, and three variants of the intruder's silhouette are shown in Figs 4 and 5.

The binary silhouettes are useful for a classification of intruders using moment-based expressions of low order [17], from which several useful descriptors of the intruders can be derived. The full-image silhouettes can be further processed for detection of interest points and subsequently for the identification of known intruders (more details to be given in subsection 3.4). The rectangular outline is a convenient presentation of the intruder for a human inspection. The visual characteristics of the intruder's image remain unchanged while the amount of data to be wirelessly transmitted is dramatically reduced.

(A) (B)

**Fig. 4. An exemplary background image and the corresponding scene containing an intruder.**

(A) (B) (C)

**Fig. 5. Intruder's silhouettes extracted in three versions.**

If a human operator supervises a large number of nodes (with the attached cameras) the chances of visual intruder detection in a single image looking like Fig. 5C are much higher then in hundreds of images looking like Fig. 4B.

### 3.2 Dynamic Intrusion Detection

Detected intruders can be further characterized by using the variations of the intruder's shape extracted from a sequence of images. This actually indicates how the intruder moves and the results can be subsequently used to classify the intruder by the type of its mobility. The example in Fig. 6 shows a pair of (overlapping) silhouettes of a human. A simple comparison of the low order moments calculated from both silhouettes, would indicate that a vertical intruder of irregular shape is moving toward the node and turning to the left. The speed of motion can be estimated based on the size changes and displacements of the gravity centre. Such a characteristic could be a sufficient evidence to recognize the intruder as a human.

In case of some man-made objects, the silhouettes extracted from the images would have more consistent shapes so that a broad classification of such objects can be done within the second-level node by using moment invariants (e. g. [17, 18]).

Generally, intrusions that can be satisfactorily identified at the second-level node are not sent for visual verification by a human operator. Unknown cases, however, have to be inspected. The images of such intruders would be wirelessly transmitted to the next level (which would usually incorporate a human operator). In order to save the bandwidth, actually only the rectangular outlines of silhouettes (e. g. Fig. 5C) are transmitted.

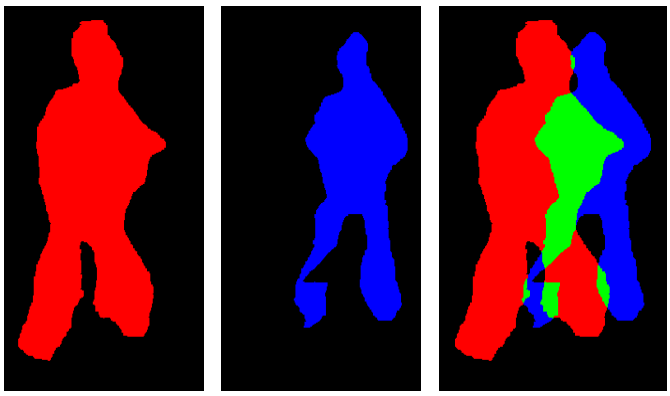


**Fig. 6. Silhouette variations in a sequence of images.**

From the Fig. 5, we can determine that the intruder is apparently approaching the camera and turning to the left. The speed of the motion can be estimated from the relative size of both silhouettes and from their relative displacement in both images.

### 3.3 Background Update

The presented system has been designed under the general assumption that the intruders can be associated with other physical phenomena, so that relatively simple sensors (proximity, vibration, magnetic, etc.) can be used as warning devices indicating a potential presence of intruders. Thus, if there is a change in the camera-captured images without the presence of the corresponding sensor warnings, it should be considered a background change rather than intrusion.

Therefore, the algorithm used for intruder extraction can also be used for periodical background update. It is particularly useful if the background change is due to rapid illumination changes (additional shadows, etc.). For slow changes of the background a simple correlation method has been implemented. When the average value of correlation falls below a certain threshold (we have experimentally verified that for night conditions the threshold should be lower than for sunny days) the background is updated. The background updating procedure can be run periodically (excluding time intervals when intrusion detection algorithm is active) with the frequency determined by both the expected dynamics of background changes and the power constraints of the second level nodes.

### 3.4 Advanced Analysis of Intruder Images

In some of the prospective applications, selected intruders might be known to the system and such should not be considered a potential danger. Such detection, therefore, should not be reported to a human operator. Unfortunately, the moment-based classification methods mentioned in the previous sub-section are not robust enough for a positive verification of known intruders. Moreover, they generally fail when intruders are only

partially visible. Therefore, selected methods originally developed for vision-based robotic navigation are being adopted for the positive verification of known intruders.

The proposed approach is based on detection and matching interest points in relative scale (e. g. [9, 19]). Interest points (sometimes referred to as corner points) are easily perceivable small areas where the *corner response* (based on the matrix of 2D partial derivatives of the image intensities–see [9]) reaches its local maximum. In the database images of known intruders, the interest points can be automatically extracted and their characteristics memorized. Examples of interest points automatically found in two images of real objects are shown in Fig. 7.

If an intruder of interest is represented by a dataset of its interest points (including geometric relations between the points) an image of such an intruder can be hypothetically identified by matching image-extracted interest points to the dataset. The major practical difficulties, i. e. sensitivity to geometric and photometric transformations (i. e. illumination variation, scale of the objects, perspective distortions of 3D objects, etc.) have been successfully solved in the algorithm presented in [9] and [20]. Exemplary results of matching interest point of a model intruder and a camera-captured image of a partially visible intruder (under different illumination conditions and in a different scale) are presented in Fig. 8. It should be noted that the matching has been done for the intruding object only partially visible within a cluttered environment. The images to be analyzed within the developed systems contain only the intruder's silhouette so that the number of interest points to be extracted and matched is relatively small.

Detection and description of interest points is a relatively simple operation that can be easily handled by the FPGA available in the developed nodes. Then, the characteristics of the points would be wirelessly transmitted to the higher level in the decision chain where the dataset of known intruders (and their interest points) is stored and the interest points can be matched, [10].

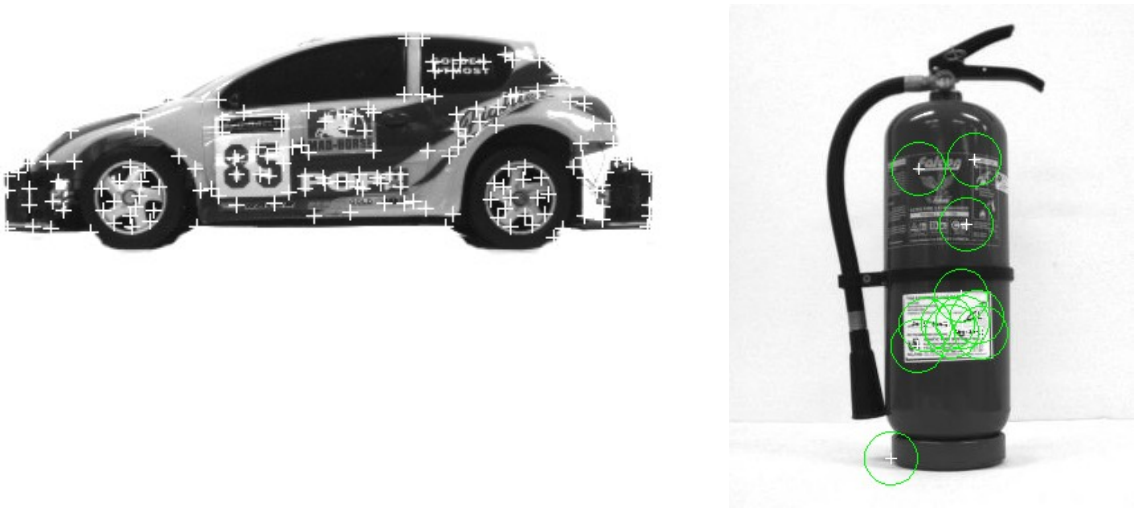


**Fig. 7. Interest points automatically extracted from images of real objects.**

**Fig. 8. Interest points automatically matched in a model image and a camera-captured view.**

This method actually reduces the amount of data transmitted in the wireless network. Within the category of known intruders the system becomes fully autonomous, but the unidentified intrusions would still be reported to the human operator, i. e. their rectangular outlines (see Fig. 5c) would be transmitted wirelessly. This method can also be applied if the second level nodes are attached to a mobile platform as it can work even if the intruders silhouette has not been extracted from the background scene.

## 4. Communication Issues

In wireless networks, the efficiency and (sometimes) security of communication is an important issue. Although within our platform no special attempts have been made to develop new communication mechanisms, the existing standards have been thoroughly tested. Additionally, we have implemented (as a feasibility study) communication between FPGA devices directly connected to a wireless transceiver (Fig. 9). It is believed that such a direct implementation of communication mechanisms within FPGA can lead to more compact and efficient nodes for the future sensor networks. More details are presented in [21].

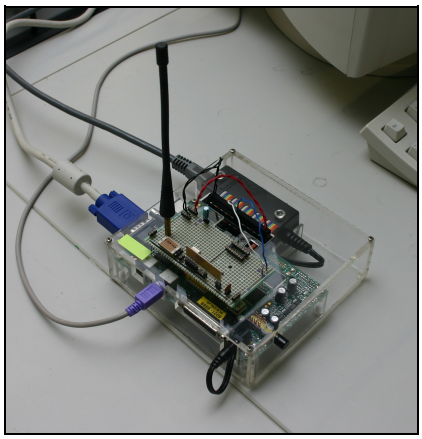


**Fig. 9. Testbed for direct wireless communication between FPGA devices.**

The most bandwidth-consuming communication task is image transfer. Thus, the amount of transmitted visual data is reduced as described in Section 3. Additionally, we envisage various compression algorithms to be locally executed before transmission. Several papers have analyzed compression standards to be used in the wireless sensor network applications for efficient power management [6, 11]. These algorithms can be designed to adopt certain parameters like compression ratio, image resolution based on the real-time situation [6, 12, 13]. The image fragments and other results are encrypted using advanced encryption standard (AES128) before the transmission to provide more security which is an important requirement for many applications.

## 5. Future Works and Conclusions

The system reported in this paper has been implemented using commercially available development boards (RC200 FPGA development boards and Ember microcontroller boards). The final prototype model will have a dedicated architecture designed based on conducted tests and further analysis and researches. Similar concepts of an FPGA-based wireless sensor network are discussed in details in [5].

In a longer term perspective, several important issues have to be carefully analyzed and eventually optimized. Adaptability of the network is one of the most important requirements. In general, the low-level sensing mechanisms are less application dependent than the data processing and interpretation algorithms. Thus, a reconfigurable FPGA has been selected for the second-level nodes that perform the complex application-dependent tasks. Simultaneously, a library of FPGA configuration files is being gradually built so that the same hardware shell can be used for a variety of tasks. Since the FPGA chip can be reconfigured wirelessly, the application of the network can changed/modified after it has been deployed. Additionally, we exploit the advantages of partial FPGA reconfigurability. By adapting this approach, we can keep permanent fragments of the tasks (e. g. standard image acquisition and preprocessing algorithms) in once-programmed structures of FPGA while the application-specific algorithms can be quickly deployed through fast partial reconfigurability.

Partial reconfigurability is a part of a more general problem of distributing the network functionalities between the programmable and fixed-logic (or once-programmable) components. While programmability (which in sensor network is almost equivalent to reconfigurability) is generally a welcome feature, it also has significant disadvantages. Higher costs of reprogrammable components and significantly higher power requirements are the most important drawbacks of FPGA-based solutions.

Additionally, we are currently studying other aspects of the future development of wireless sensor networks with vision capabilities. For example, the problem of reliable network functioning in case of a partial destruction or imperfect deployment is particularly important in certain applications. Again, it can be achieved by either increasing the number of nodes or by incorporating a higher level of reconfigurability into a node (or by a combination of both).

In this paper, we presented a two-level structure of nodes that can be used for visual intrusion detection and classification, with a preliminary support of various non-visual sensors. The FPGA based second level nodes can be dynamically configured for different applications and tasks even after the nodes are deployed using online (wirelessly) reconfiguration capabilities. Several algorithms of image processing and visual assessment implemented in FPGA have been discussed. This system can be an efficient intrusion detection system with a relatively high level of intelligence and reasonable power requirements, though still requiring a human assistance for the final decisions. Currently, a commercial product based on the developed platforms is being designed. We also believe the proposed concept of a wireless sensor network with enhanced vision capabilities could be a useful step in development of more advanced systems for man-machine collaboration.

## Acknowledgment

The authors gratefully acknowledge the support and funding from IntelliSys Research Centre, a partnership between Nanyang Technological University (Singapore) and Singapore Technologies Engineering.

## References

1. K. Obraczka, R. Manduchi, and J. J. Garcia-Luna-Aceves, *Managing the Information Flow in Visual Sensor Networks*, Proc. WPMC 2002: 5th Int. Symp. on Wireless Personal Multimedia Communication, Oct. 2002, Honolulu.
2. D. Estrin, *Sensor network research: Emerging challenges for architecture, systems, and languages*, 10th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X*)*, ACM SIGPLAN notices, vol. **37**, 10. (eds C. Norris and J. J. B. Fenwick), New York: ACM Press, Oct. 2002, pp. 1–4.
3. R. Collins, A. Lipton, and T. Kanade, *A System for Video Surveillance and Monitoring*, Proc. of the American Nuclear Society (ANS), 8th Int. Topical Meeting on Robotics and Remote Systems, Pittsburgh, Apr. 1999.
4. R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, *Algorithms for cooperative multisensor surveillance*, Proceedings of the IEEE, Vol. **89**, No. 10, Oct. 2001, pp. 1456-1477.
5. S. J. Bellis, K. Delaney, B. O'Flynn, J. Barton, K. M. Razeeb, C. O'Mathuna, *Development of Field Programmable Modular Wireless Sensor Network Nodes for Ambient Systems*, Computer Communications (accepted for special issue on Wireless Sensor Networks), to appear in 2005.
6. J. Lach, D. Evans, J. McCune, J. Brandon, *Power Efficient Adaptable Wireless Sensor Network*, *Military and Aerospace Programmable Logic Devices* Int. Conf. MAPLD 2003, Washington D. C., Sep. 2003.

7. B. Meffert, R. Blaschek, U. Knauer, R. Reulke, A. Schischmanow, F. Winkler *Monitoring traffic by optical sensors*. 2nd Int. Conf. on Intelligent Computing and Information Systems, Cairo, Mar. 2005.
8. A. E. Gamal, *Collaborative visual sensor networks*, 2004. `http://mediax.stanford.edu/projects/cvsn.html`
9. M. S. Islam, A. Sluzek and L. Zhu, *Towards invariant interest point detection of an object,* 13th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Feb. 2005.
10. M. S. Islam and L. Zhu, *Matching interest points of an object,* IEEE Int. Conf. on Image Processing ICIP2005, Genova, Sept. 2005.
11. H. Wu and A. Abouzeid, *Energy Efficient Distributed JPEG2000 Image Compression in Multihop Wireless Networks,* 4th Workshop on Applications and Services in Wireless Networks ASWN2004, Boston, Aug. 2004.
12. C. N. Taylor and S. Dey. *Adaptive image compression for enabling mobile multimedia communication*, IEEE Int. Conf. on Communications 2001, Helsinki, June 2001.
13. C. N. Taylor, S. Dey, and D. Panigrahi. *Energy/latency/image quality trade-offs in enabling mobile multimedia communication*, in: Software Radio – Technologies and Services (ed. E.D. Re), Springer Verlag, 2001, pp 55–66.
14. K. Römer and F. Mattern, *The design space of wireless sensor networks,* IEEE Wireless Communications, Vol. **11** No. 6, Dec. 2004, pp. 54-61.
15. M. A. M. Vieira, D. C. da Silva Jr., C. N. Coelho Jr., and J. M. da Mata. *Survey on Wireless Sensor Network Devices,* Emerging Technologies and Factory Automation (ETFA03). IEEE Conf., Vol. **1**, Sept. 2003.
16. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, *Coverage problems in wireless ad-hoc sensor networks*, IEEE Infocom, April 2001, pp 1380-1387.
17. M. K. Hu, *Visual pattern recognition by moment invariants,* IRE Trans.Inf.Theory vol. **8**, pp 179-187, 1962.
18. S. Maitra, S. *Moment invariants*, Proc. of IEEE, vol. **67** (4), pp. 697–699, 1979.
19. C. Schmid, R. Mohr and C. Bauckhage, *Evaluation of interest point detectors*, Int. Journal of Computer Vision*,* vol. **37** (2), 2000, pp. 151-172.
20. M. S. Islam and A. Sluzek, *Detecting and Matching Interest Point in Relative Scale*, Machine Graphics & Vision, accepted Sep. 2005.
21. G. C. S. Chen, *Wireless communication between FPGA boards* SCE04-040 FYP report, NTU, Singapore, 2005.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.19-25
ISBN 83-922646-0-6


# Experiences of Software Engineering Training

Ilona Bluemke, Anna Derezińska

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/17, 00-665 Warszawa, Poland
{I.Bluemke, A.Derezinska}@ii.pw.edu.pl

**Abstract.** In this paper some experiences from laboratories of Advanced Software Engineering course are presented. This laboratory consists of seven exercises. The subjects of exercises are following: requirements engineering, system design with UML [1], reuse, precise modeling with the Object Constraint Language (OCL [2]), code coverage testing, memory leaks detection and improving application efficiency. For each laboratory exercise a set of training materials and instructions were developed. These Internet materials are stored on a department server and are available for all students and lecturers of this course. Rational Suite tools [3] are used in the laboratory. The goal of introducing Internet materials was to improve the quality of SE education. Some experiences and observations are presented. The evaluation of student's results is also given.

## 1 Introduction

Much has been already done in the area of teaching software engineering. Every year there are dedicated international conferences on this subject, e. g. Conference on Software Engineering Education and Training, or in conferences on software engineering there are special sessions, e. g. Automated Software Engineering, Software Engineering. In proceedings from these conferences many interesting ideas or curricula's can be found, e. g. [4, 5, 6]. In this paper we present some experiences from advanced software engineering laboratory introduced in the Institute of Computer Science in fall 2004. At the Department of Electronics and Information Technology, Warsaw University of Technology two software engineering courses are taught every semester. The first one, called Software Engineering, is given to the students on the fifth semester and the second one -Advanced Software Engineering (ASE), is given to the same students on the sixth semester. During Software Engineering course students learn the basic ideas of software engineering and object modeling in UML (the Unified Modeling Language [1]). They also spend 15 hours in laboratory preparing a project of a simple system in Rational Rose. Advanced Software Engineering course should widen student's knowledge of software engineering. To improve the quality of education in ASE course the laboratory was enhanced with Internet support. Choosing subjects for ASE course, the directions from Software Engineering Body of Knowledge SWEBOK [7] were considered. It was not possible to use all the directions from SWEBOK and present all aspects of software engineering in 30 hours of lectures. The chosen subjects are the following:

- requirements engineering,
- design with reuse,
- precise modeling,
- testing,
- quality improvement,
- software effectiveness,
- software processes.

The subjects shown above relate to six out of ten Knowledge Areas (KAs) proposed by SWEBOK [7].

In ASE course students have 15 laboratory hours in blocks of two. Seven laboratory subjects, tightly connected with lectures, were prepared. At the Institute of Computer Science every semester about sixty students, divided into groups of 7-8, are having ASE laboratory. Five instructors supervise these groups. When there are several supervisors students often complain that theirs demands, requirements and grades are not the same. In order to overcome this problem detailed instructions were written for each laboratory session. While using these instructions the laboratory sessions supervised by different lecturers are similar and the time spent in laboratory can be used very effectively.

The goal of ASE laboratory was also to present the professional CASE tools supporting the different phases of software process. The tools from Rational Suite [3] used during laboratories are Requisite Pro, PureCoverage, Quantify and Purify. These tools are complicated and it is difficult to use them without training. Rational tutorials for these tools, available in English, are rather long and can not be used during two hours of laboratory. A set of tutorials for these tools was prepared in Polish, native language for the students. The tutorials and the laboratory instructions are stored on the department server and are available to students, lecturers and instructors of ASE course. Each student can study the instruction and tutoring material at home, so the instructor's directions can not surprise her/him.

In Section 2 Internet materials are briefly presented. The subjects of ASE laboratory are presented in Section 3 and illustrated with examples. Evaluation of laboratory results is discussed in Section 4.

## 2 Internet materials

The ASE laboratory materials are written in html and stored on the department server. Only students and lectures of this course have access to these materials. The structure of ASE laboratory materials is following:

- Introduction containing ASE laboratory goals
- Tutoring materials:
  - Requisite Pro
  - Code Coverage
  - Purify
  - Quantify
  - OCL
- Design Patterns
- Format of use case description
- Laboratory directions
- Vocabulary
- Bibliography.

The organization of these materials is hierarchical. At every screen there are links to the "parent" section and the main index, to the next and previous sections. In each laboratory instruction there are links to the appropriate tutoring material and positions in vocabulary (Fig. 1).

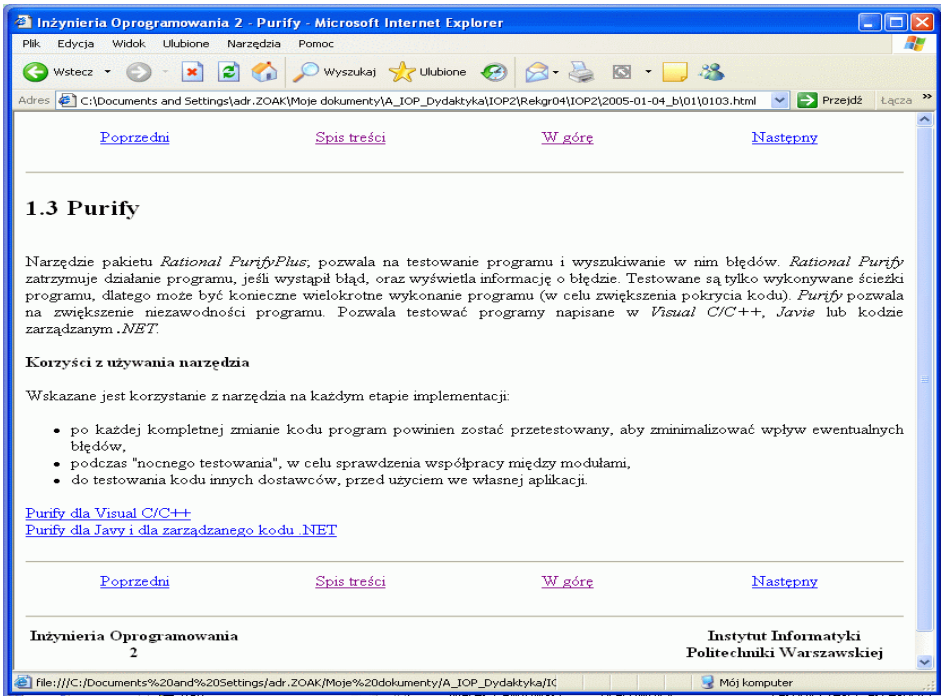



**Fig. 1.** Screen from ASE laboratory system

# 3 ASE Laboratory

The main goal of ASE laboratory was to refine student's knowledge and experience. Seven laboratory exercises were developed. The subjects of these exercises are as follows:

1. Requirements engineering
2. Software design
3. Reuse
4. Precise modeling with OCL
5. Code coverage testing
6. Quality improvement
7. Improving effectiveness.

Subjects of exercises one and three to seven are completely new to students. In all laboratory exercises the tools from Rational Suite: Requisite Pro, Rose, PureCoverage, SoDa, Purify, Quantify are used. Students on the sixth semester are familiar only with Rose and SoDa, because these tools were used in software engineering laboratory in the former semester. Rational Suite provides tutorials (in English) for most tools. Each of these tutorials needs at most two hours, so they are too long to be used during two hours in laboratory. During ASE lectures there is no time to present and discuss CASE tools in details. Problems with a tool usage could disturb the student in the development of a laboratory subject. To overcome difficulties caused by using several CASE tools, short tutorials in Polish, native language for the students, were prepared. All these tutorials are stored on the department server and are available to ASE students all semester. Although OCL and the design with reuse are presented on ASE lectures, some tutoring materials to these subjects are also available.

Students often complain that laboratories conducted by different instructors are not the same. To solve this problem detailed directions for each laboratory exercise were written and stored on the department server as well.

Each of laboratory instruction consists of three parts:

1. Introduction,
2. Detailed directions,
3. Exercise results.

In each laboratory instruction there are links to tutoring materials, the bibliography and a description of difficult ideas in the vocabulary part. An introduction describes the goals of the exercise and the basic concepts. The tasks to be prepared before a laboratory exercise can be defined in an introduction. The second part contains a sequence of tasks that should be executed during the laboratory session. Finally, the expected results and the evaluation criteria are given.

In Section 3.1 the translation of a part of the instruction "Requirements Engineering" is presented. Section 3.2 describes the organization of laboratory and in Section 3.3 discrepancies often met by the instructors are shown.

## 3.1 Example of laboratory instruction

Below, an extract from a translated instruction for the first ASE exercise, concerning requirements engineering, is presented. *Italic* font is used to denote tool's buttons, options and underline shows links to other parts of ASE internet materials, dots (...) indicate the omitted parts of the instruction:

**Introduction**

The aim of the exercise is the development of requirements specification for a given system and training in requirements management [8]. The requirements management covers the following activities: systematic eliciting, organizing and documenting requirements. In order to preserve the contract between the client and the development team, the changes in requirements must be handled. Requirements specification should follow the quality criteria (IEEE 830 [9]). The requirement's repository is created using RequisitePro, and a use case model is created using Rational Rose.

**Tasks**

1. Create a requirements repository using Requisite Pro (...)

Then, iteratively execute steps from 2 to 4.

2. Define different program features, stakeholder requests, non-functional and functional requirements Create requirements in the appropriate packages, e. g. use case. (...)
3. Specify requirements
   - Define stakeholders for stakeholder requests.

- Define requirement attributes: priority, difficulty, stability, cost.
- Create hierarchy of requirements, e. g. for requirement UC2 its descendants UC2.1 and UC2.2 (use option *New->Child Requirement*).
- Connect requirements with traceability links. Create traceability matrix. For a given package choose *New->View* and select *Traceability Matrix* in *View Type.* (...)

4. Create a use-case model
   - Create a new UML project in Rose.
   - Generate use cases and associate with the requirements of the type *Use Case* from RequisitePro (...). Adjust requirements' text with the use case names. (...)
   - Draw use-case diagram(s) in Rose. Create actors and connect them with the appropriate use cases.

Add requirements description for the use cases in *Documentation* fields in Rose. The formatted pattern of a use case description can be followed.

5. Generate documentation files (...).

**Results**

Final results of the laboratory comprise three documentation files, indicated above, and the use-case model. (...).

It should be noted that the instruction includes mainly information concerning the CASE tools and specifying the artifacts to be prepared by the students. Other activities not described in the instructions are also necessary. For example, during an exercise on requirements engineering students should negotiate the goals of his/her project with others in the same group (see Section 3.2). We assume that the instructor is responsible for stimulating these activities during laboratory exercises. It seems to us, that too long and detailed instructions are not effective.

### 3.2 Laboratory organization

The ability to work in a team is very important for a software engineer and should be trained as well. Interesting proposal how to practice the teamwork can be found for example in [10]. However developing the ASE laboratory we were aware of the significance of the teamwork, we decided to incorporate such training in this laboratory only in a very limited extent due to the time constraints (only 15 hours of laboratory). In ASE laboratory one instructor - lecturer has a group of seven to eight students. A group of students is preparing requirements for a common subject, divided into individual subtasks. The members of the group collaborate with each other defining the relations between the subtasks. To show some teamwork problems, the specification of requirements prepared during the first laboratory is reviewed by another student. The results of the reviewing process are shown in Section 3.3. On the second and the third laboratory, the reviewer designs the system using the reviewed by her/him specification. Some OCL specifications should be inserted in the project, for example in the specification of an operation. The use of design patterns is also recommended [11].

During the exercises one to three, the instructor is coming to each student and is pointing out the weak points in the design or specification. It usually takes all the time available for him/her in the laboratory and it needs a lot of effort.

Exercises five, six, seven were devoted to the code coverage testing and to the quality and effectiveness improvement. In these exercises students, grace to detailed instructions, can work more independently. Therefore these exercises can possibly be executed in a "remote" manner.

A student has to send his/her exercise results to the tutor by email. The instructor is evaluating the received artifacts (remarks, specification, projects, code, etc.) and inserts the result into the department server. At the beginning of the next laboratory session the instructor indicates the student all detected flaws. However such a laboratory evaluation is time consuming we hope it is worthy. Obtained results of ASE tests show significantly higher scores than in the previous semesters; without Internet support and with on-line exercise evaluation.

### 3.3 Discrepancies in requirements engineering

The analysis of artifacts received from students is time consuming. Many factors are relevant. Below, some factors taken into account while evaluating exercise "Requirement engineering" are listed:

- correctness, consistency and completeness of requirements set,
- logical categorization of features (goals), stakeholder requests and use-case requirements,
- specification of stakeholder requests:
  - clear and complete identification of stakeholders,
  - complete and proper assignment of stakeholder requests to stakeholders,

- identification of non-functional requirements and its specification,
- logical distribution of requirements to different levels of requirements hierarchy,
- definition of attributes of requirements:
  - assignment of requirements priorities,
  - assessment of realization difficulty,
  - recognition of stable and unstable requirements,
- clear and unambiguous requirements description,
- definition of traceability relation between requirements:
  - adequate selection of traceability relations,
  - concise directions of traceability arrows.
  - completeness of traceability relations,
- association of requirements to appropriate use- cases in use-case model,
- quality of use-case diagrams:
  - proper identification of an actor,
  - correct syntax of relations between use-cases,
  - adequate usage of different relations,
  - clear structure of use-case model,
  - readability of the diagrams,
- quality of use-case description (a detailed pattern was given in training materials):
  - identification of the realization flows and consecutive steps,
  - adequate abstraction level of step description (functional description versus an interface description),
  - clear pre- and post-conditions (if necessary).

It should be noted that our students prepared already a simple use-case model in the previous semester. In the current semester the students properly recognized the general concepts of use-cases, but still made some elementary syntax errors in the use-case diagrams.

The final verification of the requirements is a working system and the satisfaction of its users. During a teaching process the verification should be done immediately and a student should obtain a feedback about his/her work. Therefore, the results of the requirements engineering task are read by an instructor and also reviewed by another student. The following artifacts are evaluated: documentation with a description of requirements, a hierarchy and traceability relations, and a use-case model. We were surprised to find out that students were able to detect in theirs reviews many errors. The errors were often similar to the ones they made themselves in theirs own requirements specification. Below a list of flaws recognized by students while reviewing the requirements prepared by a colleague is given. Numbers, given in brackets, show in percentage in how many reviews the flaw was detected.

- Missing requirements (64%)
- Unclear or missing traceability relations (45,5%)
- Different qualification of features and use-case requirements (45,5%)
- Lack of functionality, which should be available for a recognized stakeholder/user (41%)
- Ambiguous definition of requirements (41%)
- Assignment of requirements to different modules of the system (36%)
- Incorrectly assigned priority to requirements (36%)
- Improper usage of relations between use-cases (include, extend, generalization) (32%)
- Unclear definition of stakeholder requests (32%)
- Insufficient description of a use-case, missing desired pre- and post-conditions (32%)
- Illegible use-case diagrams (23%)
- Inadequate assessment of the realization difficulty (23%)
- Missing actors in use-case model (23%)
- Lack of defined requirements attributes - importance, stability (18%)
- Incorrect hierarchy of features or use cases (9%)
- Ambiguous name of an actor (9%)
- Superfluous requirements (4,5%)

In general, students were able to correctly identify weak points in the requirement’s descriptions, even, if they made the same errors. From the list given above it can be seen that students detect many discrepancies from the list shown at the beginning of this section.

## 4 Laboratory evaluation

Objective evaluation of students' skills is very difficult and need a lot of time. In ASE laboratory a tutor has a relative small number of students (seven to eight). Evaluation of all laboratory exercises is based on:

- An activity of a student, including the approach to the solutions and presented partial results during exercises in laboratory.
- Realized tasks (specifications, models, programs) and final remarks received via email.
- A critical discussion about received results.
- A review by a colleague (only for the first exercise).

The received results contributed to the score of an exercise. Figure 2 shows the percentage of points given to students by four instructors in spring 2005. Three parts of the laboratory were evaluated independently. On average the results of the first part, the requirements engineering, and of the second part, the project with elements of reuse and OCL, were about 79%. The results of the last part concerning testing, quality and performance analysis (exercises five, six, seven - section 3) were higher (88%). The subjects of these exercises required many manipulations with the new tools (this could be easily done using the instructions and tutorials) but a less creative work than subjects of exercises one to four. Moreover, the students were more capable and willing to cope with any problems related directly to the running programs, than to the problems concerning requirements and models. The analyzed programs were not very complicated, because the main goal was to learn the new methodology.

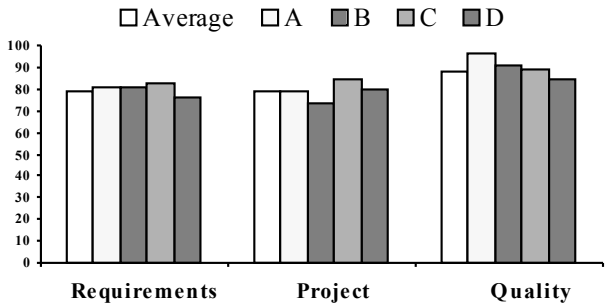



**Fig. 2.** Laboratory results

Advanced Software Engineering course is given to students of two specializations: Engineering of Information Systems (EIS) and Information and Decision Systems (IDS). All students attend the same lectures and have to pass the same exams. EIS students have ASE laboratory supported by Internet materials presented in this paper and IDS students make a project. IDS students can also use Internet materials. The laboratory and the project cover similar areas of software engineering and give the same number of credit points.

Table 1 Exam results

| Exam subjects | | Results of students | |
|---|---|---|---|
| | | after ASE laboratory | after project |
| Fall 2004/2005 | Requirements | 40% | 35% |
| | UML/OCL | 49% | 34% |
| Spring 2005 | Requirements | 93% | 81% |
| | UML/OCL | 63% | 33% |
| Sum | | 62% | 40% |

In table 1 the results of exams from two semesters are shown. The number of points obtained by students, divided by maximal amount available, is given in percentage. From the exam only the results of tasks concerning subjects practiced during the laboratory and the project were selected. Internet materials are available to all students. During lectures all students were encouraged to study these materials. In two separate columns the results of EIS students - having the laboratory are compared with the results of IDS students – having the project. It can be noticed that students having the laboratory, forced to study our Internet materials, achieved significantly higher scores.

## 5 Conclusion

The need to include a wide scope of subjects in very limited time motivated us to prepare an Internet support for the ASE course (about 90 pages). This benefited in significantly increased quality of SE education. On the exam, students attending the ASE laboratory achieved higher scores than students having projects, which also have access to these materials and even were encouraged to use them.

The ASE laboratory comprises phases of software process like the requirements engineering, system design with reuse and design patterns, precise modeling with OCL, code coverage testing, and improving the quality and effectiveness of application. Usually, these subjects are taught in separate courses and do not appear together in curricula of a single, obligatory software engineering course.

The ASE laboratory was introduced in fall 2004 and about hundred students finished it. So far it was observed that students were able to do much more than in the previous semester without Internet instructions. The demands of different instructors, placed to students, were similar. Very close to each other were also the average grades. Instructors were able to devote more attention to the design assessment and refinement; they did not waste time explaining detailed steps of each exercise.

Our intents were to use industrially proven tools in the ASE laboratory. By providing the tutoring materials a student could take an advantage of the information they provide without being exposed to the hassles of learning to use the tools. Students were satisfied working with professional CASE tools. They also get some practical experience considering software design, implementation, testing and effectiveness.

We have shown that in a short time, they can gather experiences broadening their understanding of engineering a program.

## 6 Acknowledgements

This work was supported by the Dean of the Faculty of Electronics and Information Technology, Warsaw University of Technology (grant number 503/G/1032/2900/000).

## References

1. Unified Modelling *Language Specification*, `www.omg.org/uml`
2. Warmer J., Kleppe A.: *The Object Constraint Language Precise Modeling with UML*, Addison Wesley, (1999)
3. Rational Suite, `http://www-306.ibm.com/software/ rational/`
4. Jazayeri M.: *The education of a Software Engineer*, Proceedings of the 19th Inter. Conf. on Automated Software Engineering, ASE'04, (2004).
5. Rombach D.: *Teaching how to engineer software,* Proceedings of 16th Conference on Software Engineering Education and Training, CSEE&T, Mar. 20-22, (2003)
6. Nikula U. *Experiences of embedding training in a basic requirements engineering method*, Proc of 17th Conf on Software Engineering Education and Training, CSEET'04, (2004), 104-109
7. www.swebok.org, *Guide to the Software Engineering Body of Knowledge,* (2004)
8. Leffingwell D, Widrig D.: *Managing software requirements. A unified approach*, Addison Wesley, (2000)
9. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*
10. Nawrocki J. R.: *Towards Educating Leaders of Software Teams: A New Software Engineering Programme at PUT,* in P. Klint, J. R. Nawrocki (eds.), *Software Engineering Education Symposium SEES'98* Conference Proceedings, Scientific Publishers OWN, Poznań, (1998), 149-157
11. Gamma E., Helm R., Johnson R., Vlissides J., *Design patterns: Elements of reusable object-oriented software,* Addison-Wesley, (1995).

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 27–37 

# Incremental Document Map Formation: Multi-stage Approach

Krzysztof Ciesielski, Michał Dramiński, Mieczysław A. Kłopotek,
Dariusz Czerski, Sławomir T. Wierzchoń

Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa,Poland
kciesiel,dcz,mdramins,klopotek,stw@ipipan.waw.pl

**Abstract.** The paper presents a methodology for incremental map formation in a multi-stage process of a search engine with map based user interface[1]. The architecture of the experimental system allows for comparative evaluation of different constituent technologies for various stages of the process. The quality of the map generation process has been investigated based on a number of clustering and classification measures. Some conclusions concerning the impact of various technological solutions on map quality are presented.

## 1 Introduction

Document maps become gradually more and more attractive as a way to visualize the contents of a large document collection.

The process of mapping a collection to a two-dimensional map is a complex one and involves a number of steps which may be carried out in multiple variants. In our search engine BEATCA [8–13], the mapping process consists of the following stages (see Figure 1): (1) document crawling (2) indexing (3) topic identification, (4) document grouping, (5) group-to-map transformation, (6) map region identification (7) group and region labeling (8) visualization. At each of theses stages various decisions can be made implying different views of the document map, generated by different algorithms.

For example, the indexing process involves dictionary optimization, which may reduce the documents collection dimensionality and restrict the subspace in which the original documents are placed. Topics identification establishes basic dimensions for the final map and may involve such techniques as singular value decomposition analysis (SVD [1]), fast Bayesian network learning (ETC [14]) and other. Document grouping may involve various variants of growing neural gas (GNG) techniques [5], hierarchical SOM [3] and Artificial Immune Systems [2]. The group-to-map transformation is run in BEATCA based on SOM ideas [15], but with variations concerning dynamic mixing of local and global search, based on diverse measures of local convergence [12]. The visualization involves 2D and 3D variants.

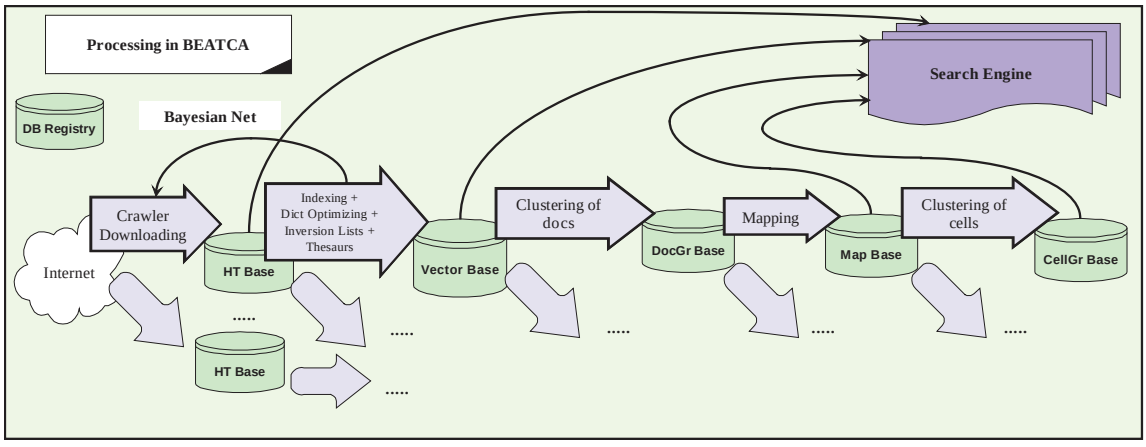



**Fig. 1.** BEATCA system architecture

[1] Research partially supported under KBN research grant 4 T11C 026 25 "Maps and intelligent navigation in WWW using Bayesian networks and artificial immune systems"

With a strongly parameterized map creation process, the user of BEATCA can accommodate map generation to his particular needs, or even generate multiple maps covering various aspects of document collection.

The overall complexity of the map creation process, resulting in long run times, as well as the need to avoid "revolutionary" changes of the image of the whole document collection, require an incremental process of accommodation of new incoming documents into the collection.

Within the BEATCA project we have devoted much effort to enable such a gradual growth. In this study, we investigate vertical (emerging new topics) and horizontal (new documents on current topics) growth of document collection and its effects on the map formation capability of the system.

To ensure intrinsic incremental formation of the map, all the computation-intense stages involved in the process of map formation (crawling, indexing and all the stages of map formation: GNG-based document grouping, model visualization and map region identification) need to be reformulated in terms of incremental growth.

In particular, Bayesian Network driven crawler is capable of collecting documents around an increasing number of distinct topics. The crawler learning process runs in a kind of horizontal growth loop while it improves its performance with increasing number of documents collected. It may also grow vertically, as the user can add new topics of for search during its run time.

The indexer has been constructed in order to achieve incremental growth and optimization of its dictionary with the growing collection of documents. Query extension capability of the query answering interface, based on Bayesian network and GNG derived dynamic automated thesaurus, accommodates also to the growing document collection. Though the actual clustering algorithms used in our system, like GNG, AIS or fuzzy C-means, are by their nature adaptive, nonetheless their tuning and modification was not a trivial task, especially with respect to our goal to achieve quality of incremental map comparable to the non-incremental one.

Special algorithms for thematic map initialization as well as for identification of document collection topics, based on GNG, SVD and/or Bayesian networks, lead to stabilization of the overall map. At the same time GNG detects the topic drift and so it may be appropriately visualized, due to plastic clustering approach, as new emerging map regions. It should be stressed at this point, that the map stabilization does not preclude obtaining different views of the same document collection. Our system permits to maintain several maps of the same document collection, obtained via different initializations of the map, and, what is more important, automatically tells the user which of the maps is most appropriate to view the results of his actual query.

In BEATCA search engine, Bayesian Networks are used in a few critical moments. We found it very useful for initial clustering of documents set. For the map creation phase, a couple of clearly separated clusters is calculated. Such clusters proved to be especially useful for thematic initialization of a clustering model and SOM projection model (the latter is shortly described in section 4).

BN is also used as thesaurus in our system. After indexing phase in BEATCA, a special dedicated BN is built on all terms in the dictionary[2]. Having collected relevant set of documents on a given subject, a joint information stored in BN and in the main clustering model will constitute context-dependent thesaurus. There is no room for the details, so we only notice that such thesaurus is used to expand user queries, for the purpose of more precise search in BEATCA search engine [13].

In the current paper we focus on our incremental version of GNG-based clustering phase of the map creation process. In section 2 we introduce the concept of GNG. In section 3—our major modification of GNG algorithm: the robust allocation of documents to clusters. In section 4 our original approach to GNG visualization is presented.

To evaluate the effectiveness of the overall incremental map formation process, we compared it to the "from scratch" map formation in our experimental section 5. A brief discussion of related works is presented in section 6. The conclusions from our research work can be found in section 7.

## 2 Growing Neural Gas approach to Clustering of Text Documents

An efficient solution to the problem of document clustering offers the Growing Neural Gas (GNG) network, first presented in [5]. Like Kohonen (SOM) networks [15], GNG can be viewed as topology learning algorithm. Its aim can be summarized as follows: given some collection of high-dimensional data, find a topological structure that closely reflects the topology of the collection. If we treat single

[2] excluding terms of low clustering quality identified during dictionary optimization phase [8]

GNG graph node as a cluster of data then the whole network can be viewed as a meta-clustering structure, where similar groups are linked together by graph edges.

In typical SOM the number of units and topology of the map is predefined. As observed in [5], the choice of SOM structure is difficult, and the need to define a decay schedule for various parameters is problematic.

GNG network starts learning with a few units[3] and new ones are inserted successively every few iterations. To determine where to insert new units, local error measures are gathered during the adaptation process; new unit is inserted near the unit which has accumulated maximal error. Interestingly, nodes of the GNG network are joined automatically by links, hence as a result a possibly disconnected graph is obtained, and its connected components can be treated as different data clusters.

The complete GNG algorithm specification and its comparison to numerous other soft competitive methods can be found in [4].

In our approach, objects (text documents as well as graph nodes, described below) are represented in the standard way, i.e. as vectors of the dimension equal to the number of distinct dictionary terms. A single element of so-called *referential vector* represents importance of a corresponding term and is calculated on the basis of the normalized TFxIDF measure. Similarity measure is defined as the cosine of the angle between corresponding vectors.

### 2.1 Utility factor

Typical problem in web mining applications is that processed data is constantly changing - some documents disappear or become obsolete, while other enter analysis. All this requires models which are able to adapt its structure quickly in response to non-stationary distribution changes. Thus, we decided to adopt and implement GNG with utility factor model [6].

A crucial concept here is to identify the least useful nodes and remove them from GNG network, enabling further node insertions in regions where they would be more necessary. The utility factor of each node reflects its contribution to the total classification error reduction. In other words, node utility is proportional to expected error growth if the particular node would have been removed. There are many possible choices for the utility factor. In our implementation, utility update rule of a winning node has been simply defined as $U_s = U_s + \text{error}_t - \text{error}_s$, where s is the index of the winning node, and t is the index of the second-best node (the one which would become the winner if the actual winning node would be non-existent). Newly inserted node utility is arbitrarily initialized to the mean of two nodes which have accumulated most of the error: $U_r = (U_u + U_v)/2$.

After utility update phase, a node $k$ with the smallest utility is removed if the fraction $\text{error}_j / U_k$ is greater then some predefined threshold; where $j$ is the node with the greatest accumulated error.

## 3 Robust winner search in GNG network

Similarly to Kohonen algorithm, most computationally demanding part of GNG algorithm is the winner search phase. Especially, in application to web documents, where both the text corpus size and the number GNG network nodes is huge, the cost of even a single global winner search phase is prohibitive.

Unfortunately, neither local-winner search method (i.e. searching through the graph edges from some staring node) nor joint-winner search method (our own approach devoted to SOM learning [8]) are directly applicable to GNG networks. The main reason for this is that a graph of GNG nodes can be unconnected. Thus, standard local-winner search approach would prevent document from shifting between separated components during the learning process.

A simple modification consist in remembering winning node for more than one connected component of the GNG graph[4] and to conduct in parallel a single local-winner search thread for each component. Obviously, it requires periodical (precisely, once for an iteration) recalculation of connected components, but this is not very expensive[5].

A special case is the possibility of a node removal. When the previous iteration's winning node for a particular document has been removed, search processes (in parallel threads) from each of its direct neighbors in the graph are activated.

[3] the initial nodes referential vectors are initialized with our broad topic initialization method, briefly described in section 4.

[4] two winners are just sufficient to overcome the problem of components separation

[5] in order of $O(V + E)$, where V is the number of nodes and E is the number of connections (graph edges)

We have implemented another method, a little more complex (both in terms of computation time and memory requirements) but, as the experiments show, more accurate. It exploits data structure known as Clustering Feature Tree [18] to group similar nodes in dense clusters. Node clusters are arranged in the hierarchy and stored in a balanced search tree. Thus, finding closest (most similar) node for a document requires $O(\log_t V)$ comparisons, where V is the number of nodes and t is the tree branching factor (refer to [18]). Amortized tree structure maintenance cost (node insertion and removal) is also proportional to $O(\log_t V)$.

## 4 Adaptive visualization of the model

Despite many advantages over SOM approach, GNG has one serious drawback: high-dimensional networks cannot be easily visualized. However, we can build Kohonen map on the referential vectors of GNG network, similarly to the case of single documents, i.e. treating each vector as a centroid representing a cluster of documents.

To obtain the visualization that singles out the main topics in the text corpus and reflects the conceptual closeness between topics, the proper initialization of SOM cells is required. We have developed special initialization method, intended to identify broad topics in the document collection and to improve the stability of the 2D/3D visualization. Briefly, in the first step the centroids of a few main clusters are identified (via fast ETC algorithm [14] and SVD decomposition [1]). Then, we select *fixpoint cells*, uniformly spread them on the map surface and initialize them with the centroid vectors. Finally, we initialize remaining map cells with *intermediate* topics, calculated as the weighted average of main topics, with the weight proportional to the Euclidean distance from the corresponding fixpoint cells and their density.

After initialization, the map is learned with the standard Kohonen algorithm [15]. Finally, we adopt so-called *plastic clustering* algorithm [17] for precise adjustation of the position of GNG model nodes on the SOM projection map, so that the distance on the map reflects as close as possible the similarity of the adjacent nodes. The concept is based on the attraction-repulsion technique, similar to the gravity clustering methods, where the nodes are attracted with the force proportional to their similarity and mass (density), while simultaneously they are repelled by graph edges. It should be stressed that the thematic initialization of the map is crucial here to assure the stability of the final visualization and to emphasize topics which are considered to be user-important [11].

The resulting map is a visualization of GNG network with the detail level depending on the SOM size (a single SOM cell can gather more than one GNG node). User can access document content via corresponding GNG node, which in turn can be accessed via SOM node-interface here is similar to the hierarchical SOM map case.

The exemplary map can be seen on Figure 2. The color brightness is related to the number of documents contained in the cell. Each cell containing at least one document is labeled with a few descriptive terms (only one can be seen on the map, the rest is available via BEATCA search engine). The black lines represents borders of thematic areas[6]. It is important to stress that this planar representation is in fact a torus surface (which can also be visualized in 3D), so the cells on the map borders are adjacent.

## 5 Experiments

To evaluate the effectiveness of the overall incremental map formation process, we compared it to the "from scratch" map formation. In this section we describe the overall experimental design, quality measures used and the results obtained.

The architecture of our system supports comparative studies of clustering methods at the various stages of the process (i.e. initial document grouping, broad topics identification, incremental clustering, model projection and visualization, identification of thematic areas on the map and its labeling). In particular, we conducted series of experiments to compare the quality and stability of GNG and SOM models for various model initialization methods, winner search methods and learning parameters [12]. In this paper we focus only on evaluation of the GNG winner search method and the quality of the resulting incremental clustering model with respect to the topic-sensitive learning approach.

[6] crisp borders are induced from fuzzy clustering of map nodes, based on the combination of Fuzzy C-Means algorithm and minimal spanning tree; algorithm details can be found in [10]

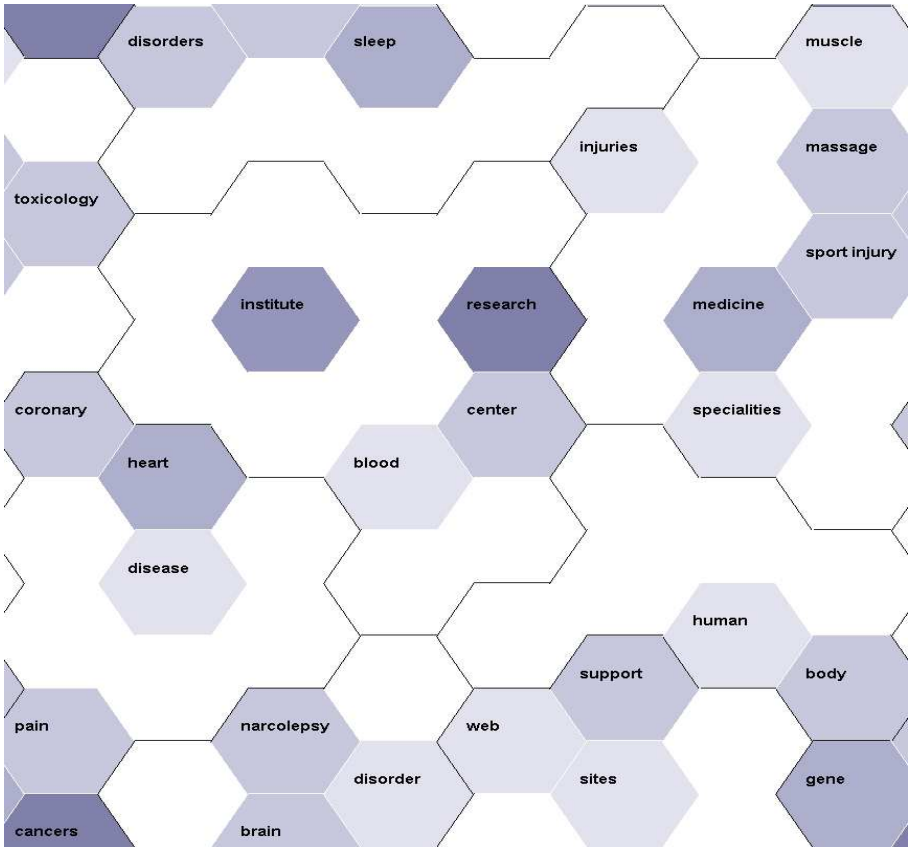



**Fig. 2.** Example of GNG model visualization for health-related newsgroups

### 5.1 Quality Measures for the Document Maps

Various measures of quality have been developed in the literature, covering diverse aspects of the clustering process (e.g. [19]). The clustering process is frequently referred as "learning without a teacher", or "unsupervised learning", and is driven by some kind of similarity measure. The term "unsupervised" is not completely reflecting the real nature of learning. In fact, the similarity measure used is not something "natural", but rather it reflects the intentions of the teacher. So we can say that clustering is a learning process with hidden learning criterion. The criterion is intended to reflect some esthetic preferences, like: uniform split into groups (topological continuity) or appropriate split of documents with known a priori categorization. As the criterion is somehow hidden, we need tests if the clustering process really fits the expectations. In particular, we have accommodated for our purposes and investigated the following well known quality measures of clustering:

- **Average Map Quantization**: the average cosine distance between each pair of adjacent nodes. The goal is to measure topological continuity of the model (the lower this value is, the more "smooth" model is):

$$AvgMapQ = \frac{1}{|N|} \sum_{n \in N} \left( \frac{1}{|E(n)|} \sum_{m \in E(n)} c(n,m) \right)$$

  where $N$ is the set of graph nodes, $E(n)$ is the set of nodes adjacent to the node $n$ and $c(n,m)$ is the cosine distance between nodes $n$ and $m$.
- **Average Document Quantization**: average distance (according to cosine measure) for the learning set between the document and the node it was classified into. The goal is to measure the quality of clustering at the level of a single node:

$$AvgDocQ = \frac{1}{|N|} \sum_{n \in N} \left( \frac{1}{|D(n)|} \sum_{d \in D(n)} c(d,n) \right)$$

  where $D(n)$ is the set of documents assigned to the node $n$.

Both measures have values in the [0,1] interval, the lower values corresponds respectively to more "smooth" inter-cluster transitions and more "compact" clusters. To some extent, optimization of one of the measures entails increase of the other one. Still, experiments [12] show that the GNG models are much more smooth than SOM maps while the clusters are of similar quality.

The two subsequent measures evaluate the agreement between the clustering and the a priori categorization of documents (i.e. particular newsgroup in case of newsgroups messages).

– **Average Weighted Cluster Purity**: average "category purity" of a node (node weight is equal to its density, i.e. the number of assigned documents):

$$AvgPurity = \frac{1}{|D|} \sum_{n \in N} \max_{c} \left( |D_c(n)| \right)$$

where $D$ is the set of all documents in the corpus and $D_c(n)$ is the set of documents from category $c$ assigned to the node $n$.

– **Normalized Mutual Information**: the quotient of the total category and the total cluster entropy to the square root of the product of category and cluster entropies for individual clusters:

$$NMI = \frac{\sum_{n \in N} \sum_{c \in C} |D_c(n)| \ \log \left( \frac{|D_c(n)| \ |D|}{|D(n)| \ |D_c|} \right)}{\sqrt{\left( \sum_{n \in N} |D(n)| \ \log \left( \frac{|D(n)|}{|D|} \right) \right) \left( \sum_{c \in C} |D_c| \ \log \left( \frac{|D_c|}{|D|} \right) \right)}}$$

where $N$ is the set of graph nodes, $D$ is the set of all documents in the corpus, $D(n)$ is the set of documents assigned to the node $n$, $D_c$ is the set of all documents from category $c$ and $D_c(n)$ is the set of documents from category $c$ assigned to the node $n$.

Again, both measures have values in the $[0, 1]$ interval. Roughly speaking, the higher the value is, the better agreement between clusters and a priori categories. At the moment, we are working on the extension of the above-mentioned measures to ones covering all aspects of the map-based model quality, i. e. similarities and interconnections between thematic groups both in the original document space and in the toroid map surface space.

### 5.2 Experimental results

Model evaluation were executed on 2054 of documents downloaded from 5 newsgroups with quite well separated main topics (antiques, computers, hockey, medicine and religion). Each GNG network has been trained for 100 iterations, with the same set of learning parameters, using previously described winner search methods.

In the main case (depicted with the black line), network has been trained on the whole set of documents. This case was the reference one for the quality measures of adaptation as well as comparison of the winner search methods.

Figure 3 presents comparison of a standard global winner search method with our own CF-tree based approach. Local search method is not taken into consideration since, as it has already been mentioned, it is completely inappropriate in case of unconnected graphs. Obviously, tree-based local method is invincible in terms of computation time. The main drawback of the global method is that it is not scalable and depends on the total number of nodes in the GNG model.

At first, the result of the quality comparison appeared to be quite surprising. On one hand, the quality was similar, on the other - global search appeared to be worse of the two (!). We have investigated it further and it turned out to be the aftermath of process divergence during the early iterations of the training process. It will be explained on the next example.

In the next experiment, in addition to the main reference case, we had another two cases. During the first 30 iterations network has been trained on 700 documents only. In one of the cases (represented by red line) documents were sampled uniformly from all five groups and in the $33^{rd}$ iteration another 700 uniformly sampled were introduced to training. After the $66^{th}$ iteration the model has been trained on the whole dataset.

In the last case (blue line) initial 700 documents were selected only from two groups. After the $33^{rd}$ iteration of training, documents from the remaining newsgroups were gradually introduced in the order of their newsgroup membership. It should be noted here that in this case we had an a priori information on

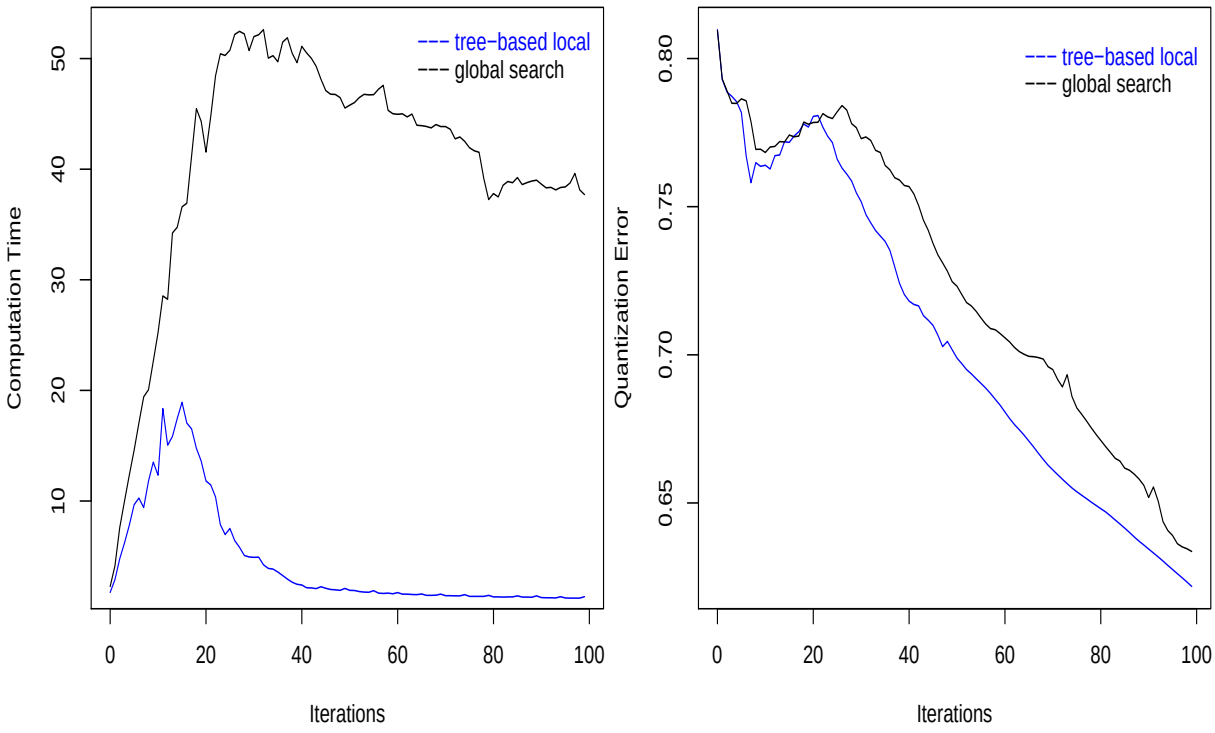



**Fig. 3.** Winner search methods (a) computation time (b) model quality

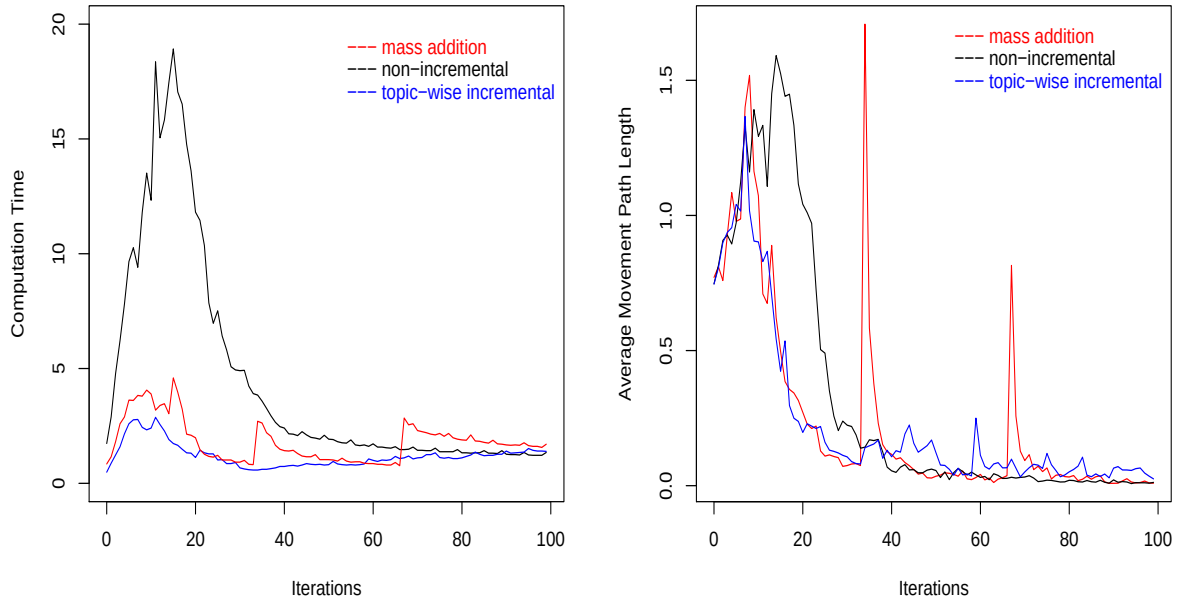



**Fig. 4.** Computation complexity (a) execution time of a single iteration (b) average path length of a document

the document category (i.e. particular newsgroup). In the general case, we are collecting fuzzy category membership information from Bayesian Net model.

As expected, in all cases GNG model adapts quite well to the topic drift. In the non-incremental and the topic-wise incremental case, the quality of the models were comparable, in terms of Average Document Quantization measure (see figure 5(a)), Average Weighted Cluster Purity, Average Cluster Entropy and Normalized Mutual Information (for the final values see table 1). Also the subjective criteria such as visualizations of both models and the identification of thematic areas on the SOM projection map were similar.

| | *Cluster Purity* | *Cluster Entropy* | *NMI* |
|---|---|---|---|
| non-incremental | 0.91387 | 0.00116 | 0.60560 |
| topic-wise incremental | 0.91825 | 0.00111 | 0.61336 |
| massive addition | 0.85596 | 0.00186 | 0.55306 |

**Table 1.** Final values of model quality measures

The results were noticeably worse for the massive addition of documents, even though all covered topics were present in the training from the very beginning and should have occupied specialized thematic areas in the model graph. However, and it can be noticed on the same plot, a complex mixture of topics can pose a serious drawback, especially in the first training iterations. In the non-incremental, reference case, the attempt to cover all topics at once leads learning process to a local minimum and to subsequent divergence (what, moreover, is quite time-consuming as one can notice on figure 4(a)). As we have previously noticed, the problem of convergence to a local minimum were even more influential in the case of global winner search (figure 3(b)).

However, when we take advantage of the incremental approach, the model ability to separate document categories is comparable for global search and CF-tree based search (Cluster Purity: 0.92232 versus 0.91825, Normalized Mutual Information: 0.61923 versus 0.61336, Average Document Quantization: 0.64012 versus 0.64211).

The figure 4(b) presents average number of GNG graph edges traversed by a document during a single training iteration. It can be seen that a massive addition causes temporal instability of the model. Also, the above mentioned attempts to cover all topics at once in case of a global model caused much slower stabilization of the model and extremely high complexity of computations (figure 4(a)). The last reason for such slow computations is the representation of the GNG model nodes. The referential vector in such node is represented as a balanced red-black tree of term weights. If a single node tries to occupy too big portion of a document-term space, too many terms appear in such tree and it becomes less sparse and—simply—bigger. On the other hand, better separation of terms which are likely to appear in various newsgroups and increasing "crispness" of thematic areas during model training leads to highly efficient computations and better models, both in terms of previously mentioned measures and subjective human reception of the results of search queries.

The last figure, 5(b), compares the change in the value of Average Map Quantization measure, reflecting "smoothness" of the model (i.e. continuous shift between related topics). In all three cases the results are almost identical. It should be noted that extremely low initial value of the Average Map Quantization is the result of the model initialization via broad topics method [8], shortly described in the section 4.

## 6 Related works

Modern man faces a rapid growth in the amount of written information. Therefore he needs a means of reducing the flow of information by concentrating on major topics in the document flow. Grouping documents based on similar contents may be helpful in this context as it provides the user with meaningful classes or clusters. Document clustering and classification techniques help significantly in organizing documents in this way. A prominent position among these techniques is taken by the WebSOM (Self

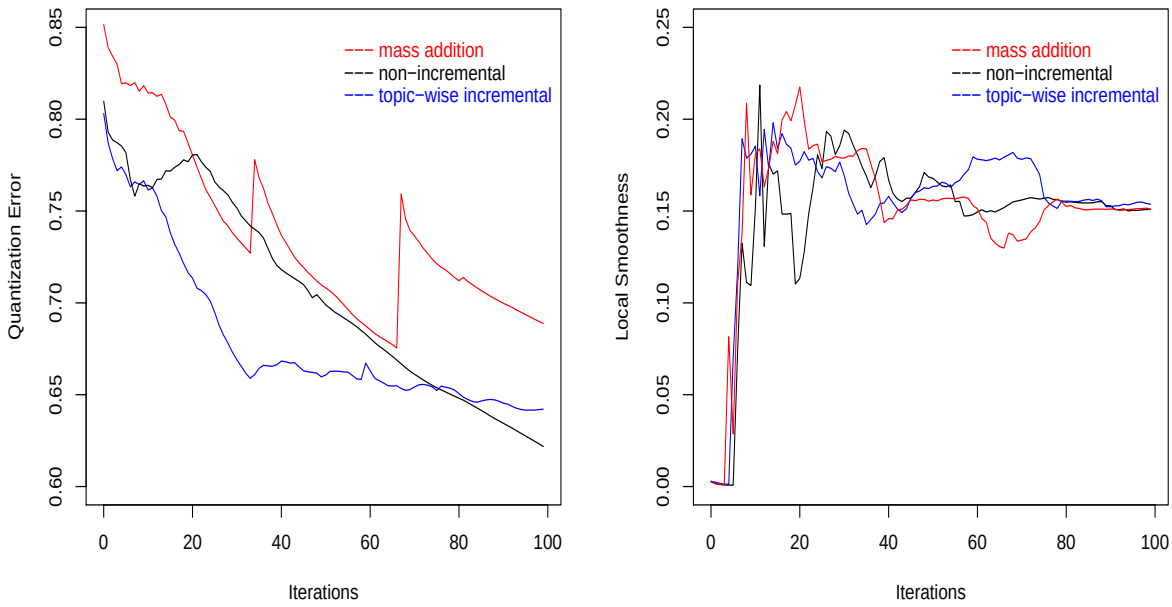



**Fig. 5.** Model quality (a) Average Document Quantization (b) Average Map Quantization

Organizing Maps) of Kohonnen and co-workers [15]. However, the overwhelming majority of the existing document clustering and classification approaches rely on the assumption that the particular structure of the currently available static document collection will not change in the future. This seems to be highly unrealistic, because both the interests of the information consumer and of the information producers change over time.

A recent study described in [7] demonstrated deficiencies of various approaches to document organization under non-stationary environment conditions of growing document quantity. The mentioned paper pointed to weaknesses among others of the original SOM approach (which itself is adaptive to some extent) and proposed a novel dynamic self-organizing neural model, so-called Dynamic Adaptive Self-Organising Hybrid (DASH) model. This model is based on an adaptive hierarchical document organization, supported by human-created concept-organization hints available in terms of WordNet.

Other strategies like that of [16, 3], attempt to capture the move of topics, enlarge dynamically the document map (by adding new cells, not necessarily on a rectangle map).

We take a different perspective in this paper claiming that the adaptive and incremental nature of a document-map-based search engine cannot be confined to the map creation stage alone and in fact engages all the preceding stages of the whole document analysis process.

## 7 Concluding remarks

As indicated e.g. in [7], most document clustering methods, including the original WebSOM, suffer from their inability to accommodate streams of new documents, especially such in which a drift, or even radical change of topic occurs.

Though one could imagine that such an accommodation could be achieved by "brute force" (learning from scratch whenever new documents arrive), but there exists a fundamental technical obstacle for a procedure like that: the processing time. But the problem is deeper and has a "second bottom": the clustering methods like those of WebSOM contain elements of randomness so that even re-clustering of the same document collection may lead to a radical change of the view of the documents. The results of this research are concerned with both aspects of adaptive clustering of documents.

The important contribution of this paper is to demonstrate, that the whole incremental machinery not only works, but it works efficiently, both in terms of computation time, model quality abd usability. For the quality measures we investigated, we found that our incremental architecture compares well to

non-incremental map learning both under scenario of "massive addition" of new documents (many new documents, not varying in their thematic structure, presented in large portions) and of scenario of "topic-wise-increment" of the collection (small document groups added, but with new emerging topics). The latter seemed to be the most tough learning process for incremental learning, but apparently the GNG application prior to WebSOM allowed for cleaner separation of new topics from ones already discovered, so that the quality (e.g. in terms of cluster purity and entropy) was higher under incremental learning than under non-incremental learning.

The experimental results indicate, that the real hard task for an incremental map creation process is a learning scenario where the documents with new thematic elements are presented in large portions. But also in this case the results proved to be satisfactory.

A separate issue is the learning speed in the context of crisp and fuzzy learning models. Apparently, separable and thematically "clean" models allow for faster learning as the referential vectors in the model nodes are smaller (contain fewer non-zero components).

From the point of view of incremental learning under soft-competitive scenario, a crucial factor for the processing time is the winner search method for assignment of documents to neurons. We were capable to elaborate a very effective method of stable, context-dependent winner search which does not deteriorate the overall quality of the final map. At the same time, it comes close to the speed of local search and is not directly dependent on the size of the model.

Our future research will concentrate on exploring further adaptive methods like artificial immune systems [2] for reliable extraction of context-dependent thesauri and adaptive parameter tuning.

## References


1. M. W. Berry, *Large scale singular value decompositions, International Journal of Supercomputer Applications,* **6**(1), 1992, pp. 13–49.
2. L. N. De Castro, F. J. von Zuben, *An evolutionary immune network for data clustering,* SBRN'2000, IEEE Computer Society Press, 2000.
3. M. Dittenbach, A. Rauber, D. Merkl, *Discovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map,* Neurocomputing, Elsevier, ISSN 0925-2312, **48** (1-4) 2002, pp. 199–216.
4. B.Fritzke, Some competitive learning methods, draft available from `http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper`.
5. B. Fritzke, *A growing neural gas network learns topologies,* in: G. Tesauro, D. S.'Touretzky, and T. K. Leen (Eds.) Advances in Neural Information Processing Systems 7, MIT Press Cambridge, MA, 1995, pp. 625–632.
6. B. Fritzke, *A self-organizing network that can follow non-stationary distributions,* in: Proceedings of the International Conference on Artificial Neural Networks '97, Springer, 1997, pp .613–618.
7. C. Hung, S. Wermter, *A Constructive and Hierarchical Self-Organising Model in A Non-Stationary Environment,* International Joint Conference in Neural Networks, 2005.
8. M. Kłopotek, M. Dramiński, K. Ciesielski, M. Kujawiak, S. T. Wierzchoń, *Mining document maps,* in Proceedings of Statistical Approaches to Web Mining Workshop (SAWM) at PKDD'04, M. Gori, M. Celi, M. Nanni eds., Pisa, 2004, pp. 87–98.
9. K. Ciesielski, M. Dramiński, M. Kłopotek, M. Kujawiak, S. Wierzchoń: *Architecture for graphical maps of Web contents,* in Proceedings of WISIS'2004, Warsaw, 2004.
10. K. Ciesielski, M. Dramiński, M. Kłopotek, M. Kujawiak, S. Wierzchoń: *Mapping document collections in non-standard geometries,* B. De Beats, R. De Caluwe, G. de Tre, J. Fodor, J. Kacprzyk, S. Zadrożny (eds): Current Issues in Data and Knowledge Engineering. Akademicka Oficyna Wydawnicza EXIT Publishing, Warszawa, 2004, pp. 122–132.
11. K. Ciesielski, M. Dramiński, M. Kłopotek, M. Kujawiak, S. T. Wierzchoń, *On some clustering algorithms for Document Maps Creation,* to appear in: Proceedings of the Intelligent Information Processing and Web Mining (IIS:IIPWM-2005), Gdansk, 2005.
12. M. Kłopotek, S. Wierzchoń, K. Ciesielski, M. Dramiński, D. Czerski, M. Kujawiak: *Understanding Nature of Map Representation of Document Collections—Map Quality Measurements,* to appear in Proceeding of International Conference on Artificial Intelligence, Siedlce, September 2005.
13. K. Ciesielski, M. Dramiński, M. Kłopotek, M. Kujawiak, S. T. Wierzchoń, *Crisp versus Fuzzy Concept Boundaries in Document Maps,* to appear in: Proceedings of DMIN-05 Workshop at The 2005 World Congress in Applied Computing, Las Vegas, 2005.
14. M. Kłopotek: *A New Bayesian Tree Learning Method with Reduced Time and Space Complexity,* Fundamenta Informaticae, **49** (no 4) 2002, IOS Press, pp. 349–367.
15. T. Kohonen, *Self-Organizing Maps,* Springer Series in Information Sciences, Vol. **30,** Springer, Berlin, Heidelberg, New York, 2001. Third Extended Edition, 501 pages. ISBN 3-540-67921-9, ISSN 0720-678X.
16. A. Rauber, *Cluster Visualization in Unsupervised Neural Networks,* Diplomarbeit, Technische Universit Wien, Austria, 1996.

17. J. Timmis, *aiVIS: Artificial Immune Network Visualization,* in: Proceedings of EuroGraphics UK 2001 Conference, Univeristy College London 2001, pp. 61–69.
18. T. Zhang, R. Ramakrishan, M. Livny, *BIRCH: Efficient Data Clustering Method for Large Databases,* in: Proceedings of ACM SIGMOD International Conference on Data Management, 1997.
19. Y. Zhao, G. Karypis, *Criterion functions for document Clustering: Experiments and analysis,* available at URL: `http://www-users.cs.umn.edu/~karypis/publications/ir.html`

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.39-47
ISBN 83-922646-0-6


# Zastosowanie algorytmu redukcji danych w uczeniu maszynowym i eksploracji danych

Ireneusz Czarnowski i Piotr Jędrzejowicz

Katedra Systemów Informacyjnych
Akademia Morska w Gdyni
Morska 83, 81-225 Gdynia
{irek, pj}@am.gdynia.pl

**Streszczenie** W pracy przedstawiono heurystyczny algorytm redukcji danych treningowych dla problemów uczenia maszynowego pod nadzorem oraz odkrywania wiedzy w oparciu o scentralizowane i rozproszone źródła danych. Proponowany algorytm wykorzystuje oryginalny mechanizm przeszukiwania wektorów uczących i wybiera wektory referencyjne tworząc zredukowany zbiór treningowy. Liczba wektorów referencyjnych zależy od wybranego przez użytkownika współczynnika poziomu reprezentacji oraz zaproponowanego w pracy współczynnika podobieństwa pomiędzy wektorami w zbiorze treningowym. Algorytm redukcji danych wykorzystuje w procesie selekcji wektorów referencyjnych algorytm uczenia populacji należący do grupy metod opartych na ewolucji populacji. W pracy przedstawiono również wyniki wybranych eksperymentów obliczeniowych.

## 1 Wstęp

Jednym z obszarów zastosowań algorytmów redukcji danych jest uczenie maszynowe. Wszystkie algorytmy uczenia maszynowego wymagają zbioru danych treningowych. Zbiór taki zawiera przypadki zwane również wektorami uczącymi, w skład których wchodzą wektory wejściowe składające się z atrybutów oraz wartości wyjściowe. Zwiększenie efektywności uczenia w tym przypadku może łączyć się z pozostawieniem w zbiorze danych treningowych tzw. wektorów referencyjnych i wyeliminowanie wektorów zawierających błędy lub szumy. Podawanie dużej ilości wektorów referencyjnych w procesie uczenia nie warunkuje wysokiej jakości klasyfikacji, a często jedynie spowalnia proces uczenia [11]. Redukcja rozmiaru zbioru treningowego prowadzi do skrócenia czasu potrzebnego na przeprowadzenie klasyfikacji oraz zmniejszenia wymagań, co do zasobów obliczeniowych. W efekcie, redukcja danych treningowych może przyspieszyć proces uczenia przy jednoczesnym zachowaniu pożądanego poziomu jakości klasyfikacji, a nawet polepszeniu jej jakości. W związku z tym uznaje się, że proces redukcji danych uczących jest istotnym elementem procesu wstępnego przetwarzania danych.

Znane algorytmy redukcji danych wybierają wektory referencyjne obliczając odległość pomiędzy wektorami w zbiorze danych treningowych. Przypadkami referencyjnymi stają się wówczas wektory leżące w okolicach centrów tworzonych przez wektory podobne. Algorytmy te wykorzystują techniki grupowania (ang.: *clustering*). Wariantem metod grupowania stosowanych w redukcji danych wejściowych jest zmniejszanie tzw. rozdzielczości danych. Inna grupa metod należąca do tzw. metod opartych na podobieństwie usuwa ze zbioru treningowego *k* najbliższych sąsiadów z danej klasy wektorów zakładając, że wszystkie wektory z sąsiedztwa będą i tak jednoznacznie klasyfikowane [9]. Istnieje jeszcze trzecia grupa metod redukcji danych. Algorytmy tej grupy eliminują wektory treningowe testując klasyfikator i redukując sukcesywnie zbiór danych wejściowych [3].

O ile stosowanie redukcji danych treningowych może przyspieszyć proces uczenia przy jednoczesnym zachowaniu pożądanego poziomu jakości klasyfikacji, o tyle żadna ze znanych metod nie gwarantuje doboru wektorów referencyjnych zmniejszającego błąd uczenia. Co więcej problem wyboru wektorów referencyjnych pozostaje ciągle aktywnym polem badań.

Drugim ważnym obszarem zastosowań dla algorytmów redukcji danych jest eksploracja danych w rozproszonych zasobach informacyjnych, a w szczególności w rozproszonych bazach danych. klasyczne podejście do eksploracji danych zakłada operowanie na danych wejściowych znajdujących się, w sensie fizycznym, w tym samym miejscu. Poważne ograniczenie dla algorytmów eksploracji danych może wynikać z naturalnego rozproszenia danych. Fizyczne rozproszenie danych jest obecnie naturalną cechą dla korporacji biznesowych,

instytucji bankowych, ubezpieczeniowych, sektora rządowego czy akademickiego. Stosowanie typowych dla scentralizowanych zbiorów danych narzędzi i algorytmów eksploracji danych nie gwarantuje identyfikacji użytecznych wzorców w środowisku rozproszonych baz danych. W przypadku heterogenicznych zbiorów danych użycie tradycyjnych metod eksploracji danych może być nawet niemożliwe [12]. Zatem odkrywanie wiedzy w oparciu o rozproszone źródła danych jest ważnym obszarem badawczym i jest postrzegane jako bardziej złożony i trudny problem niż odkrywanie wiedzy z wykorzystaniem scentralizowanych źródeł danych [10], [12].

Szeroko stosowane podejście do odkrywania wiedzy w rozproszonych zbiorach danych zakłada dwupoziomowe przetwarzanie: lokalne i globalne. Poziom lokalny, nazywany również poziomem lokalnej decyzji, dotyczy przetwarzania i eksploracji danych w miejscu fizycznej lokalizacji danych [12]. Poziom globalny dotyczy przetwarzania decyzji podejmowanych wcześniej na poziomie lokalnym.

Jedną z wyspecjalizowanych technik eksploracji rozproszonych zbiorów danych jest, tak zwane, meta-uczenie nazywane również rozproszonym uczeniem maszynowym [7]. Meta-uczenia obejmuje równoległe budowanie na poziomie lokalnym niezależnych klasyfikatorów, przy wykorzystaniu niezależnych zbiorów danych. Meta-uczenie prowadzi, na poziomie globalnym, do budowy meta-klasyfikatora integrującego modele niezależnie zbudowanych klasyfikatorów. Meta-uczenie dopuszcza stosowanie zarówno identycznych jak i różnych pod względem działania klasyfikatorów na poziomie lokalnym [4], [12].

Inne podejście do odkrywania wiedzy w rozproszonych bazach danych dopuszcza zintegrowanie wszystkich danych zawartych w niezależnych rozproszonych zbiorach danych i utworzenie dużego zbioru danych [12]. Rozszerzeniem tej koncepcji jest integrowanie, na poziomie globalnym, referencyjnych wektorów pochodzących z rozproszonych zbiorów danych. Podejście to zakłada, na poziomie lokalnym, selekcję wektorów i utworzenie zbioru reprezentatywnego, który dziedziczyłby cechy lokalnych zbiorów danych. Dla tego podejścia problem doboru odpowiedniej metody identyfikacji i selekcji wektorów referencyjnych jest problemem kluczowym [12].

W pracy zaproponowano heurystyczny algorytm redukcji danych IRA (*Instance Reduction Algorithm*) wykorzystujący metodę opartą na ewolucji populacji. Przeznaczeniem tego algorytmu jest selekcja wektorów referencyjnych i utworzenie zbioru treningowego dla algorytmu uczenia maszynowego. W pracy algorytm redukcji danych przedstawiono w dwóch obszarach zastosowań: tradycyjnego uczenia maszynowego oraz eksploracji danych w rozproszonym systemie baz danych. Ideę algorytmu IRA, oraz proponowane procedury selekcji wektorów referencyjnych oparte na wykorzystaniu algorytmu uczenia populacji oraz selekcji wektorów w rozproszonym systemie baz danych przedstawiono w części 2 pracy. Efektywność i skuteczność redukcji danych treningowych za pomocą algorytmu IRA potwierdzona eksperymentalnie. Założenia i plan przeprowadzonego eksperymentu obliczeniowego oraz uzyskane wyniki przedstawiono w części 3. Ostatnia część pracy zawiera wnioski i wskazuje dalsze kierunki badań.

## 2 Algorytm redukcji danych

### 2.1 Idea algorytmu

Pierwotnie idea algorytmu redukcji danych została przedstawiona w pracy [1]. W pracy tej algorytm redukcji danych przedstawiono jako narzędzie służące do eliminacji wektorów nadmiarowych w zbiorze treningowym przy jednoczesnym zachowaniu właściwego opisu problemu, utrzymaniu zadowalającego, poziomu jakości klasyfikacji, a w niektórych przypadkach zwiększenia jakości klasyfikacji, oraz zmniejszeniu czasu uczenia się algorytmów opartych na sztucznej sieci neuronowej.

Zadaniem algorytmu redukcji jest pozostawienie pewnej liczby przypadków z oryginalnego zbioru danych treningowych $T$ i utworzenie zredukowanego zbioru treningowego $S$. Algorytm opiera się na wykorzystaniu algorytmu uczenia populacji do wyznaczenia wektorów referencyjnych i utworzenia zredukowanego zbioru danych.

Proponowany w pracy algorytm redukcji danych przeznaczony jest do redukcji zbiorów treningowych składających się z wektorów o atrybutach typu porządkowego, liczbowego i mieszanego tj. opisanych zarówno w skali porządkowej, liczbowej jak i nominalnej. Algorytm IRA należący do klasy algorytmów tak zwanego wsadowego przeszukiwania wektorów referencyjnych (por. [11]) i wymaga wykonania trzech następujących kroków:

- obliczenie dla wszystkich wektorów z oryginalnego zbioru danych treningowego wartości współczynnika podobieństwa $I_i$,
- podział zbioru wektorów treningowych na podzbiory wektorów z identycznymi wartościami współczynnika podobieństwa,

- selekcja wektorów referencyjnych z każdego podzbioru i usunięcie pozostałych wektorów.

Niech $N$ jest liczbą przypadków w zbiorze $T$, $n$ — jest liczbą atrybutów wektora wejściowego oraz $X=\{x_{ij}\}$ (gdzie $i=1,...,N$, $j=1,...,n+1$) jest macierzą o $n$+1-kolumnach i $N$ wierszach zawierającą wszystkie wektory wejściowe wraz z wartością wyjściową z $T$ ($n$+1 element tablicy jest wartością wyjściową dla danego wektora wejściowego). Proponowany algorytm redukcji danych treningowych wykonuje pięć podstawowych etapów:

**Etap 1**: Normalizacja wartości atrybutów poszczególnych przykładów w $X$ do przedziału [0, 1] oraz zaokrąglenie ich do najbliższych wartości całkowitych.

**Etap 2**: Obliczenie dla każdego przypadku współczynnika podobieństwa $I_i$:

$$I_i=\sum_{j=1}^{n+1} x_{ij}s_j, i=1, \ldots, N, \tag{1}$$

gdzie

$$s_j=\sum_{i=1}^{N} x_{ij}, j=1, \ldots, n+1 \ . \tag{2}$$

**Etap 3**: Grupowanie wektorów z $X$ w $t$ grup $Y_v$ ($v$=1,...,$t$) zawierających wektory z identycznymi współczynnikami $I_i$, gdzie $t$ jest liczbą różnych wartości $I_i$.

**Etap 4**: Ustawienie wartości współczynnika reprezentacji $K$, który określa maksymalną liczbę wektorów uczących jaką należy zachować w każdej z $t$ grup zdefiniowanych na etapie 3.

**Etap 5**: Wybór wektorów referencyjnych i utworzenie zbioru $S$. Jeżeli przez $y_v$ oznaczymy liczbę wektorów w grupie $v$, $v$=1,...,$t$, to wybór wektorów referencyjnych przebiega następująco:

- Jeżeli $y_v \le K$ i $K>1$ to $S=S \cup Y_v$
- Jeżeli $y_v>K$ i $K=1$ to $S = S \cup \{x^v\}$, gdzie $x^v$ jest wektorem w $Y_v$, dla którego odległość $d(x^v,\mu^v)=\sqrt{\sum_{i=1}^{n}(x_i^v-\mu_i^v)^2}$ jest minimalna, a $\mu^v=\frac{1}{y_v}\sum_{j=1}^{y_v} x^v$ jest wektorem średnim w $Y_v$
- Jeżeli $y_v>K$ i $K>1$ to $S=S\cup\left\{x_j^v\right\}$, gdzie $x_j^v$ ($j$=1,...,$K$) są wektorami referencyjnymi wybranymi z $Y_v$ przez algorytm PLA.

### 2.2 Algorytm uczenia populacji

Algorytm PLA użyty do wyznaczenia wektorów referencyjnych należy do klasy algorytmów opartych na ewolucji populacji [5]. Podstawowe założenia algorytmu PLA to: populacja startowa jest dużym zbiorem dopuszczalnych rozwiązań (tzw. osobników) wygenerowanych przy wykorzystaniu wybranego mechanizmu losowego, proces uczenia populacji osobników przebiega etapowo, w kolejnych etapach używa się coraz bardziej złożonych metod uczenia (poprawy), do kolejnych etapów uczenia przechodzą osobniki spełniające kryteria selekcji. W ten sposób liczebność populacji stopniowo zmniejsza się, a najlepsze rozwiązanie na etapie finalnym traktowane jest jako rozwiązanie problemu.

W przypadku redukcji danych algorytm PLA dzieli wektory $x^v$ z $Y_v$ na $K$ podgrup $D_{vj}$, $j=1,...,K$, dla których suma kwadratów odległości euklidesowych między każdym wektorem $x^{vz}$ ($z \in D_{vj}$) i wektorem średnim $\mu^{vj}$ z $D_{vj}$ jest minimalna. Problem podziału wektorów na K podgrup związany jest z minimalizacją funkcji celu:

$$J=\sum_{j=1}^{K}\sum_{z\in D_{\text{anu }j}}(x^{vz}-\mu^{vj})^2 \tag{3}$$

Za wektory referencyjne $x_j^v$ ($j$=1,...,$K$) są obierane wektory dla których odległość do wektora średniego w danej podgrupie jest najmniejsza.

Do pozostałych założeń zaprojektowanego algorytmu PLA należą: permutacyjna reprezentacja rozwiązania, populacja startowa generowana losowo, cztery metody uczenia (losowe przeszukiwanie lokalne, krzyżowanie z częściowym odwzorowaniem PMX [6], przeszukiwanie lokalne oraz przeszukiwanie z ruchami zabronionymi - ang.: *tabu search*), wspólne kryterium selekcji (do kolejnego etapu przechodzą rozwiązania, których wartość funkcji celu jest mniejsza lub równa od średniej jej wartości w populacji).

Populacja składa się z rozwiązań o reprezentacji permutacyjnej. Każde rozwiązanie reprezentowane jest przez $K+y_v$ elementów. $K$ pierwszych pozycji określa ile z $y_v$ kolejnych elementów należy do $K$-tej podgrupy. $K$ pierwszych pozycji nie może również przyjmować wartości zero a $y_v$ kolejnych liczb reprezentuje numer wektora z $Y_v$.

Na pierwszym etapie każde rozwiązanie populacji poddawane jest poprawie z użyciem operatora losowego przeszukiwania lokalnego. Zaprojektowana metoda z losowo wybranej podgrupy wybiera losowo numer wektora i przydziela go do innej losowo wybranej podgrupy w danym rozwiązaniu. Jeśli wartość funkcji $J$ nowego otrzymanego rozwiązania jest mniejsza od jej wartości obliczonej dla poprawianego rozwiązania to nowe rozwiązanie jest akceptowane i zastępuje rozwiązanie poprawiane, w przeciwnym razie jest odrzucane. Zaakceptowanie nowego rozwiązania wiąże się również z uaktualnieniem liczebności wektorów w poszczególnych podgrupach.

Druga metoda poprawy wykorzystuje mechanizm krzyżowania z częściowym odwzorowaniem (PMX) [6]. Poprawiane rozwiązanie populacji jest krzyżowane z innym rozwiązaniem populacji wybranym z wykorzystaniem mechanizmu losowego. Jeśli wartość funkcji $J$ któregoś z dwóch potomków otrzymanych w drodze krzyżowania jest mniejsza od jej wartości obliczonej dla poprawianego rozwiązania to potomek ten zastępuje rozwiązanie poprawiane. W przeciwnym wypadku potomek wykazujący lepsze przystosowanie poddawany jest poprawie przez działanie operatora przeszukiwania lokalnego.

Operator przeszukiwania lokalnego dla losowo wybranego elementu (tj. numeru wektora) z rozwiązania poprawianego oblicza jego odległość euklidesową do wszystkich wektorów średnich pozostałych podgrup, a następnie przydziela go do podgrupy, gdzie ta odległość jest najmniejsza. Jeśli wartość funkcji $J$ tak zmodyfikowanego rozwiązania ulega zmniejszeniu to rozwiązanie to jest akceptowane wraz z uaktualnieniem liczebności wektorów w poszczególnych podgrupach, w przeciwnym przypadku jest odrzucane.

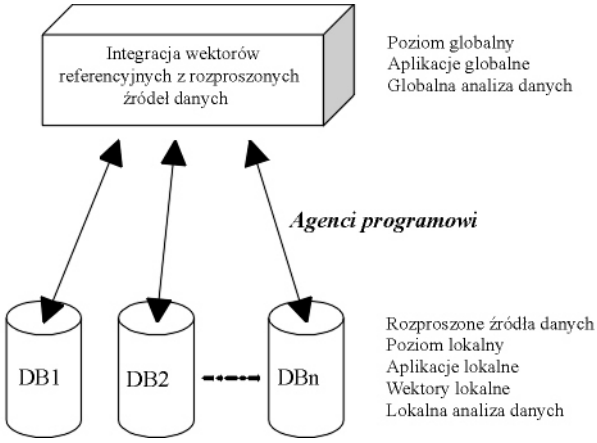
Czwarty z operatorów poprawy wykorzystuje mechanizm przeszukiwania z ruchami zabronionymi operując pamięcią ruchów zabronionych *SM*. W metodzie tej losowo wybrany numer wektora $p_i^j$ ($j$=1...$y_v$) rozwiązania $i$ populacji $P=\left\{p_i\right\}$ ($i$=1,...,$M$), gdzie $M$ jest wielkością populacji, jeśli nie należy do *SM*, jest przydzielany kolejno do wszystkich pozostałych podgrup. Jeśli przydzielenie wektora do innej podgrupy daje zmniejszenie wartości funkcji $J$ rozwiązania $i$, to zmodyfikowane rozwiązanie jest akceptowane, jeśli nie, to procedurę przyporządkowania do poszczególnych podgrup przeprowadza się dla wektora $p_i^{j-1}$ a następnie dla $p_i^{j+1}$. Jeśli zmiana przynależności do podgrup dla wektorów odpowiednio: $p_i^j$, $p_i^{j-1}$ i $p_i^{j+1}$ nie przyniesie poprawy jakości rozwiązania to numer wektora $p_i^j$ zostaje umieszczony w pamięci *SM* i pozostaje w niej przez $s$ iteracji. Wszystkie opisane procedury na poszczególnych etapach algorytmu PLA poprawiają każde rozwiązanie populacji $c$ krotnie, gdzie $c$ jest liczbą iteracji dla procedur poprawy.

### 2.3 Selekcja wektorów referencyjnych w rozproszonym systemie baz danych

Zastosowanie algorytmu IRA do selekcji wektorów referencyjnych w rozproszonym systemie baz danych opiera się na dwupoziomowym przetwarzaniu danych. Jest to typowe podejście do odkrywania wiedzy w rozproszonych zbiorach danych [12]. Algorytmu IRA w rozproszonym systemie baz danych wymaga wykonania dwóch kroków. Pierwszy krok wykonywany jest na poziomie lokalnym i dotyczy selekcji wektorów referencyjnych oraz utworzenia reprezentatywnych zbiorów danych. Drugi krok dotyczy zintegrowania wektorów referencyjnych, które zostały wyselekcjonowane na poziomie lokalnym. Krok ten odbywa się na etapie globalnym przetwarzania.

Fizyczna implementacja dwupoziomowego systemu przetwarzania danych opiera się na systemie wieloagentowym [13]. W takim systemie selekcja wektorów referencyjnych realizowany jest przez agentów programowych o kodzie źródłowym takim jak algorytm IRA. System również opiera się na protokole komunikacyjnym związanym z przesyłaniem zbiorów reprezentatywnych z poziomu lokalnego przetwarzania na poziom globalny. Na poziomie globalnym następuje integracja wektorów oraz rozpoczyna się globalna

analiza danych realizowana w oparciu o narzędzia uczenia maszynowego. Architekturę systemu dla problemu odkrywania wiedzy w rozproszonym systemie baz danych przedstawiono na Rysunku 1.




**Rysunek 1**. Architektura dwupoziomowego przetwarzania danych w rozproszonym systemie baz danych

## 3 Eksperyment obliczeniowy

Celem przeprowadzonego eksperymentu obliczeniowego było porównanie jakości klasyfikacji uzyskanej przez uczenie klasyfikatora zredukowanym zbiorem treningowych oraz z użyciem oryginalnego, pełnego zbioru treningowego. Eksperymenty obliczeniowe zostały przeprowadzone dla dwóch przypadków redukcji zbioru treningowego, tj. dla przypadku ze scentralizowaną oraz rozproszoną bazę danych. W oparciu na otrzymanych wynikach przeprowadzono analizę wpływu redukcji przykładów uczących na jakość uczenia klasyfikatora. Jako klasyfikator wykorzystano algorytm C 4.5 [8].

Eksperymenty przeprowadzono dla danych dotyczących oceny zdolności kredytowej. Problem oceny zdolności kredytowej klienta (ang.: *The Customer Intelligence in The Banking*) był przedmiotem konkursu ogłoszonego w 2002 roku w ramach projektu EUNITE - *EUropean Network on Intelligent TEchnologies for Smart Adaptive Systems* [2]. Poszczególne przykłady opisują zdolność kredytową klienta, która jest oznaczona jako *active* lub *non-active*. Dane dwuklasowego problemu składają się ze zbioru 24000 przykładów. Każdy przykład opisany jest 36 cechami o wartościach rzeczywistych, całkowitych i binarnych.

Eksperymenty obliczeniowe przeprowadzono w oparciu o test 10 - *krotnej walidacji skrośnej*. W przypadku badania wpływu redukcji danych zapisanych w scentralizowanym zbiorze danych na jakość klasyfikacji zbiór danych podzielony został na 10 równych części a algorytm IRA stosowano do redukcji zbioru treningowego składającego się z 9 części, z których następnie wygenerowano zredukowany zbiór treningowy stanowiący wejście dla wybranego algorytmu uczenia maszynowego. Pozostała 10-ta część posłużyła do testowania algorytmu uczenia. Następnie przeprowadzono 10-krotną ocenę działania klasyfikatora z wykorzystaniem 10 par zredukowanych zbiorów treningowych i testowych. Ostatecznie wyznaczono średnią trafności klasyfikowania. Przyjęty sposób weryfikacji wpływu redukcji przykładów na jakość klasyfikacji przedstawiono w pracy [11].

W przypadku eksploracji danych w oparciu o rozproszone źródła danych zbiór 24000 przykładów podzielono losowo na zbiór treningowy i testowy zawierające odpowiednio 22000 i 2000 przykładów. Następnie, tak jak ma to miejsce w przypadku rozproszonych źródeł danych, zbiór treningowy podzielono, z wykorzystaniem mechanizmu losowego, na trzy niezależne zbiory danych. W kolejnym kroku każdy z podzbiorów był poddany redukcji z użyciem algorytmu IRA. Ostatecznie wektory referencyjne z każdego z podzbiorów zostały połączone i utworzono zbiór treningowy dla algorytmu uczenia maszynowego. Test powtórzono dziesięciokrotnie dla różnych podziałów na zbiór treningowy i testowy, oraz trzykrotnie dla różnych podziałów na zbiory systemu rozproszonego wynoszących odpowiednio (5415, 11150, 5435), (6430, 9210, 6360) i (7010, 9630, 5360) elementów.

Wszystkie eksperymenty przeprowadzono dla dziesięciu różnych wartości współczynnika reprezentacji $K=\{1, 5, 10, 15, 20, 25, 30, 100, 150, 200\}$. Wyniki eksperymentów obliczeniowych przedstawiono w Tabelach 1 oraz 2. Podane wielkości stanowią wartość uśrednią po wszystkich przeprowadzonych przebiegach eks-

perymentu obliczeniowego. Wartości te zostały obliczone dla trzech wariantów algorytmu C 4.5: bez przycinania drzewa, z przycinaniem drzewa oraz z przycinaniem redukującym błąd.

Tabela 1 przestawia jakość klasyfikacji algorytmu C 4.5 uczonego zredukowanym zbiorem danych oraz przy wykorzystaniu pełnego zbioru danych. Wyniki podane w Tabeli 1 dotyczą przypadku ze scentralizowaną bazą danych. Jakość klasyfikacji dla różnych wartości współczynnika reprezentacji przedstawiono również w Tabeli 2, wyniki te jednak dotyczą przypadku z rozproszonymi źródłami danych. Dodatkowo porównanie jakości klasyfikacji dla C 4.5 z przycinaniem drzewa przedstawiono na Rysunku 2.

**Tabela 1**. Średnia jakość klasyfikacji (w %) algorytmu C 4.5 dla przypadku ze scentralizowanym źródłem danycc

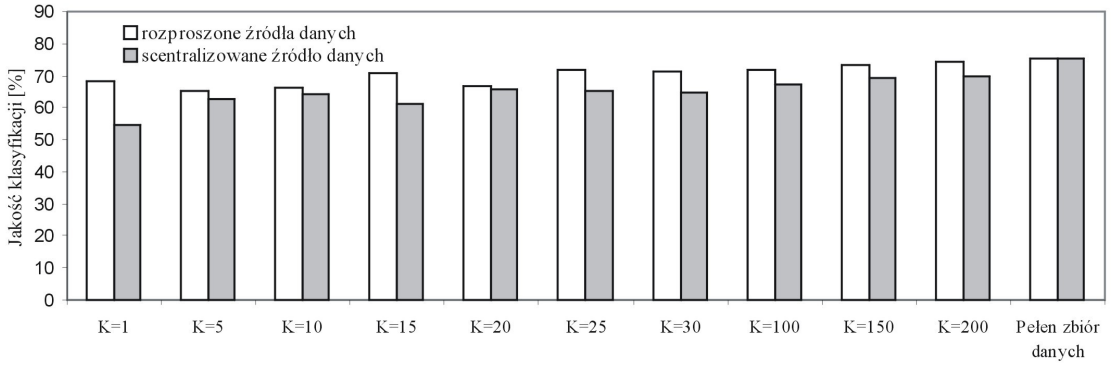
| | Wartość współczynnika poziomu reprezentacji | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | K=1 | K =5 | K=10 | K=15 | K=20 | K=25 | K=30 | K=100 | K=150 | K=200 | Pełen zbiór danych |
| brak przycinania | 54.85 | 63.7 | 63.95 | 63.55 | 65.4 | 65.75 | 64.7 | 67.4 | 66.8 | 66.8 | 73.25 |
| przycinanie | 54.85 | 62.45 | 64.15 | 61.35 | 65.6 | 65.28 | 64.85 | 67.25 | 69.05 | 69.6 | 75.5 |
| przycinanie redukujące błąd | 61.85 | 64.15 | 64.55 | 65.4 | 63.1 | 64.15 | 61.5 | 69.65 | 66.55 | 70.6 | 75.15 |

**Tabela 2**. Średnia jakość klasyfikacji (w %) algorytmu C 4.5 dla przypadku z rozproszonymi źródłami danych

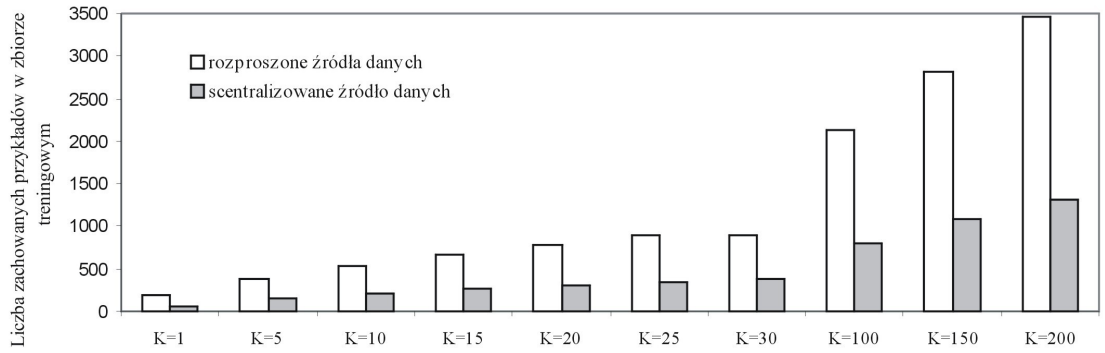
| | Wartość współczynnika poziomu reprezentacji | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | K=1 | K=5 | K=10 | K=15 | K=20 | K=25 | K=30 | K=100 | K=150 | K=200 | Pełen zbiór danych |
| brak przycinania | 67.35 | 62 | 65.05 | 68.35 | 65 | 70 | 68.75 | 67.3 | 69.55 | 70.95 | 73.25 |
| przycinanie | 68.2 | 65 | 66.2 | 70.54 | 66.75 | 71.6 | 71.5 | 71.7 | 73.5 | 74.55 | 75.5 |
| przycinanie redukujące błąd | 58.95 | 67 | 65.35 | 68.65 | 64.55 | 71.05 | 66.8 | 70.65 | 73.45 | 72.55 | 75.15 |

Uzyskane wyniki pokazują, że selekcja wektorów referencyjnych, niezależnie od fizycznej lokalizacji danych, gwarantuje uzyskanie zadowalających rezultatów uczenia klasyfikatora. Dla przykładu, dla współczynnika reprezentacji równego 10 algorytm C 4.5 z przycinaniem jest w stanie zapewnić jakość klasyfikacji na poziomie 64.15% i 66.2% odpowiednio dla przetwarzania danych scentralizowanej bazy danych i rozproszonej bazy danych. Dla współczynnika reprezentacji równego 200 jakość klasyfikacji, dla przetwarzania rozproszonych źródeł danych, wynosi 74.55%. W przypadku scentralizowanego źródła danych jest jakość klasyfikacji jest równa 69.6%. Dla porównania jakość klasyfikacji oparta na pełnym zbiorze treningowym wynosi 75.5%.

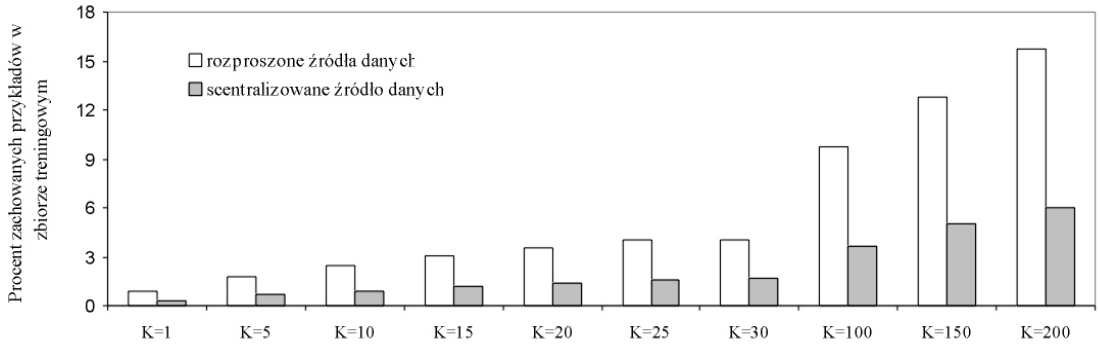
Dodatkowym elementem niezbędnym do porównania otrzymanych wyników jest liczba zachowanych przez algorytm IRA wektorów w zbiorze uczącym. Liczbę zachowanych przykładów oraz, odpowiednio, procent zachowanych przykładów w zbiorze treningowym przedstawiono na Rysunku 3 i 4. Dla przykładu, dla współczynnika reprezentacji równego 10 liczba zachowanych przykładów w zbiorze treningowym wynosi 205 i 533 dla przetwarzania danych scentralizowanej bazy danych i rozproszonej bazy danych, co stanowi odpowiednio 0.93% i 2.42%. W przypadku współczynnika reprezentacji równego 200, liczby te wynoszą 1322 i 3462 dla obu przypadków, co stanowi odpowiednio 6.01% i 15.74%.



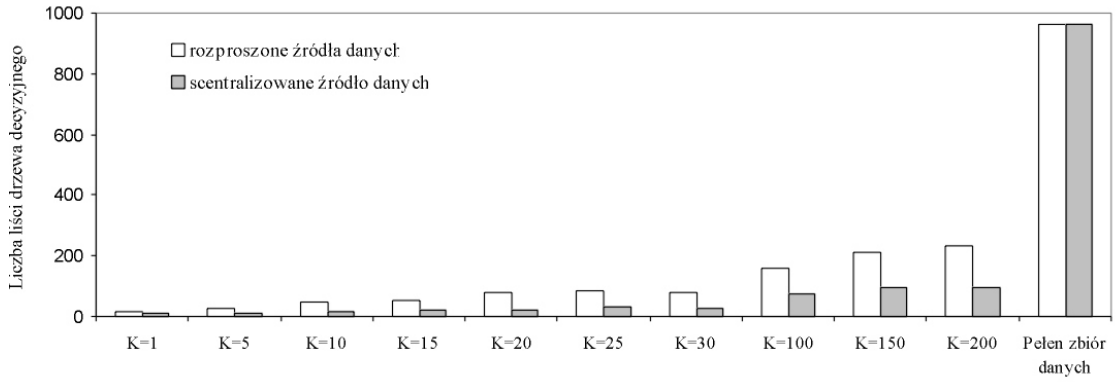

**Rysunek 2**. Porównanie jakości klasyfikacji algorytmu C 4.5 dla różnych współczynników reprezentacji




**Rysunek 3**. Liczba zachowanych wektorów w zbiorze uczącym dla różnych współczynników reprezentacji




**Rysunek 4.** Procent zachowanych wektorów w zbiorze uczącym dla różnych współczynników reprezentacji




**Rysunek 5.** Liczba liści drzewa decyzyjnego

Dodatkowo na Rysunku 5 przedstawiono liczbę liści drzewa decyzyjnego w zależności od wartości współczynnika reprezentacji. Porównanie to dotyczy zarówno przetwarzania scentralizowanych jak i rozproszonych danych oraz dotyczy algorytmu z przycinaniem drzewa. Dla przykładu, dla współczynnika reprezentacji równego 200 liczba liści drzewa decyzyjnego wyniosła odpowiednio 96 i 234, co świadczy o znacznie mniejszej złożoności struktury drzewa w porównaniu do sytuacji budowania drzewa decyzyjnego w oparciu o pełen zbiór danych, gdzie liczba liści wyniosła 964.

## 4. Zakończenie

W pracy przedstawiono heurystyczny algorytm redukcji danych dla potrzeb uczenia maszynowego oraz eksploracji danych w oparciu o scentralizowany i rozproszony system baz danych. Eksperymenty obliczeniowe pokazały, że użycie algorytmu IRA może przyczynić się do otrzymania jakości klasyfikacji nieznacznie różniącej się od tej, jaką można uzyskać wykorzystując, do budowy klasyfikatora, pełnego zbioru danych.

Eksperymenty obliczeniowe pokazały również, że reprezentacja wiedzy o klasyfikacji jest mniej złożona w przypadku, gdy jest ona budowana w oparciu o zredukowany zbiór danych niż, gdy opiera się ona na pełnym zbiorze danych treningowych. Mniej złożona reprezentacja wiedzy gwarantuje jej czytelność oraz jest korzystna z obliczeniowego punktu widzenia. W ogólności wniosek ten jest prawdziwy zarówno dla drzew decyzyjnych jak i dla większości metod reprezentacji wiedzy.

Wektory referencyjne mogą być również gromadzone we wspólny reprezentatywny zbiór treningowy dla narzędzi eksploracji danych, który, jak potwierdziły eksperymenty obliczeniowe, dziedziczy podstawowe cechy rozproszonych źródeł danych. Algorytm IRA jest w stanie wskazać istotne informacje w niezależnych zbiorach danych gwarantując tym samym wysoką jakość klasyfikacji na poziomie globalnym.

Eksperymenty obliczeniowe pokazały również, że w niektórych przypadkach podział zbioru danych na niezależne podzbiory, następnie redukcja rozmiarów tych podzbiorów i integracja wektorów referencyjnych może poprawić jakość eksploracji w porównaniu do tradycyjnych podejść. Wniosek ten może sugerować nowe podejście w eksploracji danych oparte na zasadzie dekompozycji i scalania.

Do kierunków dalszych badań należeć będzie wskazanie reguły definiowania współczynnika reprezentacji w algorytmie IRA. Dalsze badania obejmą również weryfikację innych narzędzi uczenia maszynowego pod kontem eksploracji danych w rozproszonym systemie baz danych.

## Bibliografia

1. Czarnowski I., Jędrzejowicz, P.: *An Approach to Instance* Reduction *in Supervised Learning.* In: Coenen F., Preece A. and Macintosh A. (ed.): Research and Development in Intelligent Systems XX. Springer, London (2004) 267-282
2. *The European Network of Excellence on Intelligent* Technologies *for Smart Adaptive Systems* (EUNITE) – EUNITE World competition in domain of Intelligent Technologies – `http://neuron.tuke.sk/competition2/`
3. Grudziński K., Duch W.: *SBL-PM: Simple Algorithm for* Selection *of Reference Instances in Similarity Based Methods*. In: Proceedings of the Intelligent Information Systems. Bystra, Poland (2000) 99-107

4. Hillol Kargupta, Byung-Hoon Park, Daryl Hershberger, Erik Johnson: *Collective Data Mining: A New Perspective Toward Distributed Data Analysis.* In: H. Kargupta and P. Chan (ed.): Accepted in the Advances in Distributed Data Mining, AAAI/MIT Press (1999).
5. Jędrzejowicz P.: *Social Learning Algorithm as a Tool for Solving Some Difficult Scheduling Problems.* Foundation of Computing and Decision Sciences, **24** (1999) 51-66.
6. Michalewicz Z.: *Algorytmy genetyczne + struktury danych = programowanie ewolucyjne.* Wydawnictwo Naukowo-Techniczne, Warszawa (1999).
7. Prodromidis A., Chan P. K., Stolfo S. J.: *Meta-learning in Distributed Data Mining Systems: Issues and Approaches.* In: H. Kargupta and P. Chan (ed.): Book on Advances in Distributed and Parallel Knowledge Discovery. AAAI/MIT Press (2000).
8. Quinlan, J. R.: *Improved Use of Continuous Attributes* in C 4.5. Journal of Artificial Intelligence Research **4** (1996) 77-90.
9. Salzberg S.: *A Nearest Hyperrectangle Learning Method.* Machine Learning, **6** (1991) 277-309.
10. Shichao Ahang, Xindong Wu, Chengqi Zhang *Multi-Database Mining.* IEEE Computational Intelligence Bulletin, Vol .**2**, No. 1 (2003).
11. Wilson D. R., Martinez T. R.: *Reduction Techniques for Instance-based Learning Algorithm.* In: Machine Learning. Kluwer Academic Publishers, Boston, 33-3 (2000) 257-286.
12. Xiao-Feng Zhang, Chank-Man Lam, William K. Cheung: *Mining Local Data Sources For Learning Global Cluster Model Via Local Model Exchange*. IEEE Intelligence Informatics Bulletin, 4, no. 2 (2004).
13. Yutao Guo, Jőrg P. Műller: *Multiagent Collaborative Learning for Distributed Business Systems.*In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04). IEEE Press, New York (2004).

XXI Autunm Meeting of Polish Information Processing Society
Conference Proceedings, pp.49-57
ISBN 83-922646-0-6


# Specification of Dependency Areas in UML Designs

Anna Derezińska

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19
00-665 Warsaw, Poland
A.Derezinska@ii.pw.edu.pl

**Abstract.** A concept of dependency areas can help in tracing an impact of artifacts of a project (requirements, elements of the UML design, extracts of the code) and assist in their evolution. The dependency area of an element of a UML design is a part of the design that is highly influenced by the given initial element. Dependency areas are identified using sets of propagation rules and strategies. Selection strategies control application of many, possible rules. Bounding strategies limit the number of elements assigned to the areas. This paper is devoted to the specification of the rules and strategies. They are specified using an extended UML meta-model and expressions in the Object Constraint Language (OCL).

## 1 Introduction

The Unified Modeling Language (UML)[14] is intended as a language to be used for model-driven development. A model gives the ability to consistently show different views of the same design. The views are expressed by the relevant diagrams of the UML. As the UML is semantically rich, we can widely describe the system that will be developed, but we cannot guarantee the consistency of the designed model.

There are three natures of checking the rightness of user diagrams, namely completeness, consistency and correctness [8,10,11]. The completeness states whether the user requirements are completely reflected on diagrams of a model. The consistency is responsible for checking whether the diagrams are coherently designed with only one requirement. It encompasses so-called vertical consistency that confirms the relation between corresponding models of different abstraction levels (inter-model consistency). Finally, the correctness decides whether the user diagrams are compliant to the syntactic and semantic rules of the UML standard. Within a model it is also named horizontal or intra-model consistency. In some cases it is impossible to conclude whether diagrams are inconsistent or incomplete.

The industrial projects are often incomplete [12]. They comprise mostly class diagrams. Their state diagrams are assigned only for selected classes. Such projects can have redundant or contradictory information. Traceability information can be partially missing or not up to dated. The development tools for UML models can check the syntax of the diagrams, but the consistency verification is still unsatisfactory. A systematic modeling strategy, e.g. RUP (*Rational Unified Process*) [9], can recommend an appropriate structure of a model. However its successful application depends on the effort and experience of a developer of the model.

Addressing the problems of comprehension and evolution of UML designs, a framework for the identification of dependency areas was introduced [4]. The dependency area of an initial element is a part of the model consisting of elements that are highly related to, and influenced by, this element (or a set of initial elements).

The identification of the dependency areas is not a final goal, but one step of a model-driven development. Dependency between parts of design artifacts is an issue of very high practical relevance. The approach of dependency areas can be beneficially applied in the following domains:

- Support understanding of a design.
- Ensure that requirements are correctly implemented.
- Determine the effects of parts of the specification
- Monitor the changes in the design.
- Assist in testing - assuring that an adequate part of the design was covered by the tests.

Basing on the dependency areas, defined in a design by a prototype tool, we selected the appropriate parts of the code generated from the design [4]. The automatically selected code was used in the further evaluation of the program dependability.

Dependency areas are identified using the sets of propagation rules and strategies. Selection strategies control application of many, possible rules. Bounding strategies limit the number of elements assigned to the areas.

The approach deals with imperfect designs, incomplete or inconsistent, in a tool-supportable way. Conflicting and ambiguous situations are resolved according to given strategies. The usage of the rules can be restricted by the strategies. In the strategies, the consistency with the UML specification is preferred. In some ambiguous cases the strategies are based on heuristics that try to reveal the intensions of a developer.

The focus of this paper is on the precise definition of the rules and strategies using a UML meta-model combined with a textual specification written in the Object Constraint Language (OCL) [17]. The OCL, a generic query language is a part of the current standard of modeling, the UML [14].

The paper is structured as follows. Section 2 presents an overlook of a concept of dependency areas and its framework. The specification of propagation rules and strategies is explained in the next sections. Section 5 gives basic information about experiments and a tool support. Remarks about related work and conclusions finish the paper.

## 2 Dependency areas

A concept of dependency areas is a successor of the idea of a dependency region [4]. It was later revised and specified as an extension of the UML meta-model [5]. The term area replaced the former term region, because the notion of a region is used in UML 2.0 for structuralizing of state machines [14].

An outlook of the idea of dependency areas is given below. The framework for the identification of dependency areas is briefly reminded in Section 2.2.

### 2.1 A concept of dependency areas

The ***potential dependency area*** of an initial element *a* - **PDA(*a*)** - is the set of all possible model elements accessible from *x* through relations available in the design: traceability, dependency, containment, etc.

The ***dependency area*** of an initial element *a* - **DA(*a*)** - is a certain subset of PDA(*a*). It is obtained by reducing the potential dependency area according to given strategies.

One can also define the dependency area of a given set of initial elements.

The identification of the dependency area of a given initial element is based on three general strategies:

I) Propagation of relations.
II) Selection of relations.
III) Bounding potential dependency areas.

The first strategy explores different relations between elements of a design. It decides about assigning elements to potential dependency areas. The propagation strategy can be expressed by a set of propagation rules. A propagation rule states that if an element *x* of a design belongs to a current PDA(*a*) and other pre-conditions of the rule are satisfied then selected elements of this design are also included in the PDA(*a*).

Different traceability relations can be identified for an element of a design. Therefore, the element can satisfy the pre-conditions of the different propagation rules. Moreover, some of the relations can be incomplete, ambiguous, or even contradictory. According to a selection strategy (II), we choose which of the propagation rules should be applied in the defined cases.

Many elements of a design can be assigned to a certain PDA(*a*). In an extreme case, even all elements of the design can belong to the same PDA(*a*). Such an area will be not useful for the applications of dependency areas, mentioned in the previous section. Therefore, a potential dependency area can be reduced according to a given bounding strategy (III).

The idea of dependency areas was adopted for UML designs [4, 5]. It was especially aimed at imperfect designs. A given UML design could be incomplete or the information from different diagrams is inconsistent. Therefore, the selection and bounding strategies are based on some heuristics that presume the intentions of a design developer. Using these strategies the identification of dependency areas can be performed automatically. The identification of the dependency area of an element *a* - **DA(*a*)** (i.e. its quality) can be verified by the following conditions:

- all elements of DA(*a*) are directly or indirectly dependant on the element *a* assuming the defined types of dependency,
- no element of DA(*a*) is superfluous - i.e. not substantially dependant on the element *a*.

### 2.2 Meta-model of dependency areas

A framework for identification of dependency areas in UML designs was defined using a conceptual UML model [5]. The model can be interpreted on a meta-model level - as an extension of the UML meta-model [14]. The meta-model for identification of dependency areas will be called briefly DA meta-model. A small part of the DA meta-model is shown in Fig. 1.

A potential dependency area is defined for a given element of the type *InitialElement*. The identification of potential dependency areas assigned to the initial element is controlled by *TraceabilityStrategies.* A traceability strategy consists of a set of *PropagationRules*. The class *DependencyArea* is a specialization of the class *PotentialDependencyArea.*

Any area consists of elements of the type *AreaMember*. The abstract type *AreaMember* combines two concrete types of elements *TraceTerminator* and *MetamodelTraceLinker*. These elements are derived from various elements of the UML meta-model.

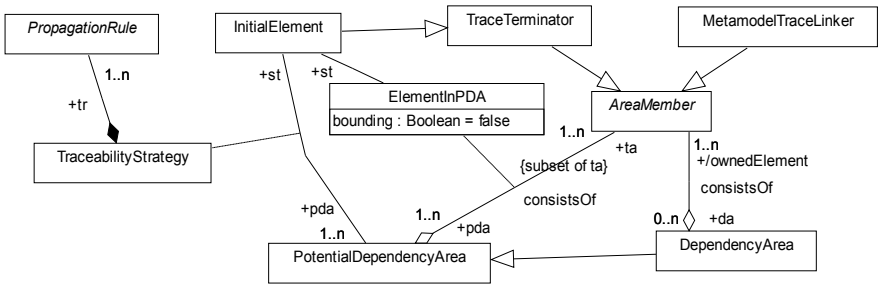



**Fig. 1.** A part of the DA meta-model - dependency areas

## 3 Propagation rules

The strategies of the framework are defined using sets of rules. The rules are specified with the notation of the Object Constraint Language (OCL)[17], which is a part of the UML standard. In this paper, we focus on the way of specification of the rules and strategies defined for the identification of dependency areas. The approach will be illustrated by simple examples. The complete mapping of the DA meta-model to the whole UML specification will be not given, for the brevity reasons.

A propagation strategy consists of propagation rules. A propagation rule can determine that different elements of the design are included to a current *PDA(a)*. These elements can be for example: properties of a class that belongs to the *PDA(a)*, a use case included in another use case from the *PDA(a)*, a whole diagram assigned to an element belonging to the *PDA(a)*. Above examples correspond to the relations within a UML model. Other propagation rules are defined also using different relations between the models (trace, access, usage etc.).

This section explains how the propagation rules are specified. First, a simple example of an OCL specification of a rule is shown. Then, the same example is defined more precisely in the context of the DA meta-model.

### 3.1 Specification of propagation rules in the OCL

Propagation rules are defined as pre and post-conditions of an operation. The conditions are written in the OCL. The following example explains an idea of a propagation rule. The reference to the UML meta-model is illustrated by an appropriate extract from the meta-model diagram [14].

The example considers a relation of the generalization between two elements of a model. Let us assume that an element *x* belongs to a certain PDA*(a)*. The element *x* is connected with the generalization *y* to its base element *z*. The assignment of the specific element (here *x*) to PDA*(a)* implies the assignment of a more general element (here z) to the same area. The considered element *x* can be for example a use case, a class or a signal.

The rule R1 is based on the part of the UML meta-model shown in Fig. 2. The generalization *z* has its corresponding class in the meta-model, similarly as the elements *x* and y. Therefore, the generalization *z* is also included in the same potential dependency area PDA*(a)*.

```
Rule R1   pre:    tr.st.pda.ta -> includes(x) and
                  x.oclIsKindOf(Classifier) and
                  x.generalization -> includes(z) and
                  z.general = y

          post:   tr.st.pda.ta -> includes(y) and
                  tr.st.pda.ta -> includes(z)
```

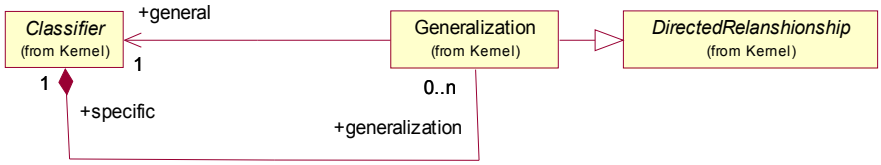



**Fig. 2.** A part of the UML meta-model -- a relation of the generalization

The pre- and post-conditions are interpreted using the combined domain of elements from the conceptual model and the UML meta-model. The OCL expression *tr.st.pda.ta* denotes the set of elements comprised in a potential dependency area of an initial element. It is specified by the names of the roles from the classes of the meta-model (Fig. 1). The navigation from the *PropagationRule* approaches the *TraceabilityStrategy* (the role *tr)*, then the *InitialElement* (the role *st*) and its *PotentialDependencyArea* (*pda*), and finally the elements of this area (*ta*). The standard OCL operation *A->includes(a)* denotes that an element *a* is contained in the set defined by the expression *A*.

### 3.2 Propagation rules in the DA meta-model

An abstract class of the type *PropagationRule* is the base class of all propagation rules in the DA meta-model (Fig. 3). Each propagation rule has its identifier and a priority. Pre- and post-conditions of a rule are specified in the context of its operation *ruleBody()*. The operation takes as a parameter an element of the type *TraceTerminator*. Any *TraceTerminator* is either the initial element of a potential dependency area or an element already included in the current area. A *TraceTerminator* represents an element from the considered UML design (the application model). Therefore, it is derived from a certain element of the UML meta-model.

In the DA meta-model a group of special classes represents a set of propagation rules. The basic idea of these classes will be explained with an example. It considers the same relation of the generalization between classifiers (Fig. 2). A part of the DA meta-model specifying this relation is shown in Fig. 3. A class of the type *ClassifierPropagationRule* is responsible for the realization of all rules devoted to classifiers. This fact is defined in the pre-condition of the operation of this class.

For a given type of a *TraceTerminator* (here a classifier) different specific rules can be defined. Therefore, other classes are aggregated to the class that determines a given type of a *TraceTerminator*. In Fig. 3 the class

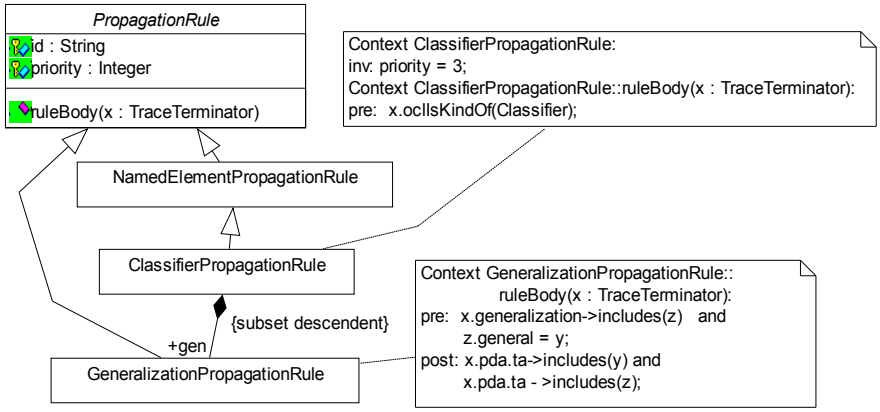



**Fig. 3.** A part of the DA meta-model - a generalization of classifiers

*CassifierPropagationRule* aggregates the class *GeneralizationPropagationRule*. Pre- and post-conditions of the operation of the class *GeneralizationPropagationRule* specify the propagation rule based on a relation of the generalization.

Consequently, analyzing a generalization of a classifier, the pre- and post-conditions from the both detailed rules, the *ClassifierPropagationRule* and the *GeneralizationPropagationRule*, should be satisfied. The logical conjunction of the conditions defined in these rules is equivalent to the conditions presented in the Rule R1 in the previous section.

Many other propagation rules are specified in the similar way in the DA meta-model.

## 4 Specification of the strategies

Specifying the selection and bounding strategies, we use a notion of the propagation rules written in the OCL. The strategies are based on heuristics that presume the intentions of the developer. The application of the rules is controlled by their position in the DA meta-model, by their priorities and additional attributes of the strategies.

### 4.1 Selection strategy

Many propagation rules can be applied to a given element (*TraceTerminator*) belonging to a potential dependency area. The application of propagation rules is controlled by selection strategies. These strategies are responsible also for the resolving of ambiguous situations. A selection strategy defines a hierarchy of applicable propagation rules. The strategy takes into account the priorities of the propagation rules and additional selection priorities, if necessary to decide among the rules of the same priority.

In the DA meta-model the propagation rules create a hierarchy. This hierarchy corresponds to the inheritance hierarchy of the base types of *TraceTerminators*, i.e. the inheritance hierarchy of elements in the UML meta-model. For example, a class of the type *ClassifierPropagationRule* is derived from the *NamedElementPropagationRule* (Fig. 3), because the class *Classifier* derives from the class *NamedElement* in the UML specification. Other propagation rules that are derived from the *ClassifierPropagationRule* exist also in the DA meta-model, e.g. a rule of a use case or a rule of a class.

A given *TraceTerminator* can satisfy pre-conditions of many propagation rules in its inheritance path of the hierarchy. In the first step, the most specific propagation rule that corresponds to a given *TraceTerminator* type will be taken into account. After considering this rule and the rules aggregated to it, its base rule, defined for a more general *TraceTerminator*, can be evaluated. For example, a *TraceTerminator* is at first recognized as a use case and later as a classifier.

Furthermore, a certain propagation rule PR corresponding to a given type of a *TraceTerminator* can have more detailed propagation rules that are aggregated to PR. It can have more then one such rule. These rules have priorities of the same or different values. At first, the rules with the highest priority are applied. Among the rules with the same priority all of them are used or only a subset can be chosen. The application of these rules is controlled by appropriate attributes of the selection strategy.

As an example, the UML dependency relation between two elements of a design is defined by the rule R2 given below. It is one of the rules defined for a *TraceTerminator* that is the *NamedElement* from the UML meta-model. A similar rule is also defined for another *TraceTerminator* - the class. According to the selection strategy and its hierarchy, this similar rule will be applied first than the rule R2.

```
Rule R2   pre:    tr.st.pda.ta -> includes(x) and
                  x.oclIsKindOf(NamedElement) and
                  x.suplierDependency-> includes(z) and
                  z.stereotype.name->includes('derive')
                  and  z.client->includes(y)

          post:   tr.st.pda.ta -> includes(y) and
                  tr.st.pda.ta -> includes(z)
```

In the rule R2, the dependency relation is specified with the stereotype «derive». Another rule can describe the same dependency relation but without any stereotype. This rule has a lower priority than the rule R2.

In some designs, no direct relations between elements can be found and/or no meaningful stereotypes are present. In this case, the rules based on an identity or a similarity of the names of appropriate elements can be

used. For example, a name of a use case is equal to a name of a collaboration. Such rules have lower priorities than the rules discussed above.

### 4.2 Bounding strategy

Bounding strategies limit a number of elements included in a potential dependency area. According to given heuristics, they try to select only those elements which are directly pointed out or intently used by a developer. Instead of identifying potential dependency areas and further excluding the assigned elements, a bounding strategy can be specified as a specific selection strategy. Bounding strategies use the similar notion of propagation rules. Within a bounding strategy the application of the rules can be restricted. The potential dependency area PDA(*a*) obtained finally according to the selection and bounding strategies is the resulting dependency area DA(*a*) identified for the initial element *a*.

Bounding strategies are based on heuristics. One of them assumes that information included in the behavioral diagrams of a design is superior to information from the structural diagrams. A simple example will illustrate this approach and its specification.

There are two diagrams concerning a class *C* in a UML design. The class *C* is defined with its operations *O1* and *O2* in a class diagram - structural diagram (Fig. 4a). The second one is a behavioral diagram assigned to the class *C* (Fig. 4b).

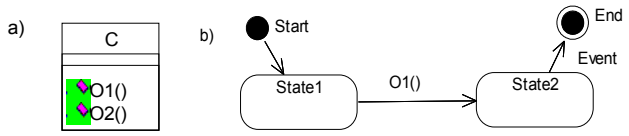



**Fig. 4. a)** A structural diagram - class C **b)** A behavioral diagram - state machine of the class C

The propagation rule R3 given below takes into account information derived from a class diagram. According to the rule R3, if the class *C* belongs to a certain PDA(*a*) then both its operations *O1* and *O2* would be assigned to the same PDA(*a*).

```
Rule R3   pre:    tr.st.pda.ta -> includes(x) and
                  x.oclIsKindOf(Class) and
                  x.ownedOperation->includes(y)

          post:   tr.st.pda.ta -> includes(y)
```

Other rules are dealing with information defined in behavioral diagrams. The class *C* is a classifier that has its behavior specification expressed by a state machine. One of transitions of this state machine is triggered by the operation *O1*. If the class *C* belongs to the PDA(*a*) a sequence of propagation rules can be performed. The last rule of this sequence is the rule R4 shown below. Appling this rule, the operation *O1* will be assigned to the PDA(*a*).

```
Rule R4   pre:    tr.st.pda.ta -> includes(x) and
                  x.oclIsKindOf(transition) and
                  x.trigger.operation->includes(y)

          post:   tr.st.pda.ta -> includes(y)
```

Without using any bounding strategy, both operations *O1* and *O2* are contained in the considered PDA(*a*). However, the operation *O2* is present only in the structural diagram. It is not used in any behavioral diagram. In this design, only the operation *O1* expresses an activity of the class C. Therefore, after using the bounding strategy, the operation *O1* will be included in the PDA(a) and the operation *O2* will be not included. The developer of the design can be warned that the operation *O2* could be missing in behavioral diagrams due to incompleteness or inconsistency of the design.

The bounding strategy is realized by the modification of the propagation rules and the restrict order of their application. As an example, the modified parts of the rules R3 and R4 are shown below. The inclusion of an element to a given area is controlled by a *bounding* attribute defined for the bounding strategy in the DA metamodel (Fig. 1). The bounding strategy assures also that the rule R4 will be applied before the rule R3.

```
Rule R3   (The modified pre-condition)
          pre:    tr.st.pda.ta -> includes(x) and
                  x.oclIsKindOf(Class) and
```

```
                x.ownedOperation->includes(y) and
                x.ownedOperation-> forAll (p|
                p.st.pda.elementInPDA.bounding = false)

Rule R4   (The modified post-condition)
          post:   tr.st.pda.ta -> exist (p| p = y and
                  p.st.pda.elementInPDA.bounding)
```

In another example, a design contains only one diagram dealing with the class *C*, namely a class diagram (Fig. 4a). In this case no information about the behavior can be used. The rule R4 is not applied and no bounding will be provided. According to the modified rule R3 both operations *O1* and *O2* are included in the PDA(*a*).

Summing up, the following heuristics is specified by the rules discussed above. If a class has no information about its behavior all its operations, given in structural diagrams, are assigned to a PDA(*a*). Otherwise, the PDA(*a*) comprises only these operations that were specified in the behavioral diagrams. The bounding strategy deals in the similar way with other members of a class and with other classifiers.

## 5 Experimental evaluation of dependency areas

The proposed strategies were empirically verified on several small UML projects. Dependency areas identified automatically by a prototype tool [4] were compared with the corresponding parts of the model selected manually by developers of the projects. In some projects the dependency areas generated by the tool contained fewer elements then areas anticipated by the developers. In an extreme case, the dependency area of a given use case was even the empty set. The analysis of such projects revealed the lack of traceability notions in them and/or a disordered structure of these projects. In general a dependency area can help in pointing out an ill-quality of a project.

In other projects, evaluated in these experiments, elements automatically assigned to dependency areas were consistent with the expectations of the developer. These areas contained all elements responsible for the realization of initial use cases. At the same time, no superfluous elements were assigned to these areas.

The analysis of the adequacy of dependency areas identified by the tool and the evaluation of the quality of a project could be supported by quantitative measures. Identification of a dependency area that has no elements or a relatively small number of them can be a warning of an incomplete design. On the other hand, a dependency area containing many, or even all, elements can indicate a project that is coupled too strongly. The analysis of a project can be also supported by other metrics, based on the number and kinds of the rules used during the identification of dependency areas. The detailed definition of all these metrics and their interpretation is beyond the scope of this paper.

The experiments, mentioned above, were performed using the prototype tool for **I**dentification of **D**ependency **A**rea -- so-called IDA. Only the elements belonging to class diagrams were taken into account during identification of dependency areas in a UML model. Next, these dependency areas were used for selection of the corresponding fragments of the C++ code generated from the model. IDA traced the changes of requirements managed by IBM-Rational RequisitePro [15]. UML models were analyzed in cooperation with the UML designer IBM-Rational Rose [15]. IDA recognized use cases assigned to the changed requirements and identified their dependency areas. All elements belonging to the dependency areas were marked with additional stereotypes, modifying the given model. A dependency area of any initial element (here a use case) was denoted with its own stereotype. In this way one element could have been assigned to many different dependency areas, eg. DA*(a)* and DA*(b)* for $a \neq b$. Dependency areas of the sets of initial elements were also identified, if required.

After analyzing the results of the experiments, we started developing an improved version of the tool. It is based on the given meta-model and the specification of pre- and post-conditions in the OCL. A new version of IDA identifies dependency areas using information from all types of UML diagrams. A dependency area can contain any element of the design. The tool supports UML 2.0.

## 6 Related work

Traceability from software artifacts (requirements, models) through to program code supports understanding the code, and how the former implement the latter [1,7,13]. Many traceability approaches address a granularity level of diagrams, not interfering into the UML details. In [13], the elements of a UML model are integrated with textual specifications of requirements, but the approach does not explore dependencies within the model.

In [16] uncertain relations between requirements and elements of a UML model are resolved by a statistical evaluation of the developers' decisions.

Our work relates also to the field of consistency of UML models. The consistency problems in UML designs were extensively studied in many papers [2,6,8,10-12]. Nevertheless, many existing approaches, especially formally-based ones, require complete traceability to guide consistency checking. Egyed [6] maintains the consistency of UML class diagrams during refinement and takes into account partial lack of traceability information. Hypothesizing that at least one of the potentially many choices ought to be consistent, the maximum flow algorithm finds the unique correct interpretation and inconsistent traces are removed.

OCL is primarily used for two purposes, for a precise description of UML models and for the specification of the UML with the relation to its meta-model [14,17]. It was successfully applied in different UML-based specifications [2,3,8,10,11]. There are some tools that offer verification of OCL constrains or animation of UML/OCL specifications [3, 11]. However, the widespread adoption of the OCL by industry is still limited due to the insufficient support and integration with other CASE tools.

## 7 Conclusions

This paper presents the specification of dependency areas identified in UML designs. The approach is based on the strategies and propagation rules. The rules are precisely defined using pre- and post-conditions denoted in the Object Constraint Language (OCL) [17]. The OCL expressions are interpreted in a meta-model (DA meta-model). It extends the UML meta-model [14] with a conceptual model of dependency areas. The rules and strategies take into account real-world designs, possibly incomplete or inconsistent.

The concept of dependency areas provides a basis for controlling changes within a design, as well as an impact of requirements on the design and the generated code. It can support automation of project management and maintenance.

Currently, the work is focused on developing a new version of IDA, a tool that can support experiments with more comprehensive projects. Furthermore, an integration of the tool with other CASE-tools is needed.

Our future work is dedicated also to the construction of certain metrics. The metrics are based on the size of dependency areas and kinds of the rules used for the identification of the dependency areas. These metrics help in assessing a quality of dependency areas identified automatically by the tool and a quality of the current stage of the design. The new tool should support the calculation of the metrics.

## Acknowledgments

I am very thankful to my student J. Zawłocki for his help in developing the IDA tool. This work was supported by the Dean of the Faculty of Electronics and Information Technology, Warsaw University of Technology, under grant number 503/G/1032/2900/000.

## References

1. Alves-Foss J., Conte de Leon D., Oman P.: *Experiments in the Use of XML to Enhance Traceability Between Object-Oriented Design Specifications and Source Code*, In: Proc of the 35th Hawaii Intern. Conf. on System Sciences, HICSS-35'02 (2002)
2. Briand L. C., Labiche Y., O'Sullivan L.: *Impact Analysis and Change Management of UML Models*, In: Proc. of International Conference on Software Maintenance, Amsterdam, The Netherlands, IEEE CS Press (2003) 256-265
3. Correa A.L. Werner C. M. L.: *Precise Specification and Validation of Transactional Business Software*, In: Proc. of the 12th IEEE Inter. Requirements Eng. Conf. (RE'04) (2004)
4. Derezińska A.: *Reasoning about Traceability* in Imperfect UML Projects, Foundations of Computing and Decision Sciences, Vol. 29, No. 1-2 (2004) 43-58
5. Derezińska A., Bluemke I.: *A Framework for Identification of Dependency Areas in UML Designs*, In: Proceedings of IASTED Inter. Conf. on Software Engineering and Application (SEA'05), Phoenix, USA, Nov. (2005) to appear
6. Egyed A.: *Consistent Adaptation and Evolution of Class Diagrams during Refinement*, In: Proceedings of the 7th International Conference on Fundamental Approaches to Software Engineering (FASE), Barcelona, Spain, March (2004) 37-53
7. Egyed A.: *A Scenario-Driven Approach to Trace Dependency Analysis*, IEEE Trans. on Software Engineering, vol. **29**, no 2 (2003) 116-132.

8. Ha L-K., Kang B-W.: *Meta-Validation of UML Structural Diagrams and Behavioral Diagrams with Consistency Rules,* In: Proc of IEEE Pacific Rim Conf on Communications, Computers and Signal Processing, PACRIM,Vol. 2., 28-30 Aug. (2003) 679-683
9. Krutchen P.: *The Rational Unified Process An Introduction,* Addison-Wesley, (2000)
10. Kuzniarz L et al. (eds.): *2nd Intern. Workshop on Consistency Problems in UML-based Software Development,* <<*UML*>> 2003, San Francisco, 20 Oct. (2003)
11. Kuzniarz L et al. (eds.): *3rd Intern. Workshop on Consistency Problems in UML-based Software Development,* <<*UML*>> 2004, Lisbon, 11 Oct. (2004)
12. Lange C. F. J., Chaudron M.R.V.: *An Empirical Assessment of Completeness in UML Designs.* In: Proceedings of "EASE–International Conference on Empirical Assessment in Software Engineering", Edinburgh, Scotland, May (2004)
13. Letelier P.: *A Framework for Requirements Traceability in UML-based Projects,* In: Proc. of 1st Int. Workshop on Traceability in Emerging Forms of Soft. Eng. by IEEE Conf. on Automated Software Engineering (ASE 02), Sept. 28, Edinburg, (2002)
14. *Object Management Group, UML 2.0 Superstructure Specification,* formal/05-07-04, www.uml.org
15. Rational Rose, Rational Requisite Pro: http://www-306.ibm.com/software/rational/
16. Spanoudakis G. et al.: *Rule-based Generation of Requirements Traceability Relations,* In: J. Systems and Software, vol. 72, no. 2 (2004) 105–127
17. Warmer J., Kleppe A.: *The Object Constraint Language: Getting Your Models ready for MDA,* Addison Wesley (2003)

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.59-71
ISBN 83-922646-0-6


# Model mapowania aktywności i kompetencji w projektach IKT

Kazimierz Frączkowski

Instytut Informatyki Stosowanej, Politechnika Wrocławska
Ul. Wybrzeże Wyspiańskiego 27
50-370 Wrocław
e-mail: kazimierz.fraczkowski@pwr.wroc.pl

**Streszczenie**: Praca dotyczy propozycji metodyki zwiększania efektywności wytwarzania projektów informatycznych a głownie skrócenie czasu ich realizacji, zmniejszenia kosztów oraz minimalizacji ryzyka poprzez dekomponowanie projektu na aktywności w postaci ich macierzy. Przewidywany do realizacji zespół projektowy podlega procesowi identyfikacji poprzez opis posiadanych kompetencji w postaci tzw. macierzy kompetencji. PM aby ocenić czy proponowany zespół będzie w stanie zrealizować projekt z powodzeniem dokonuje mapowania planowanych aktywności projektowych z kompetencjami i brakujące lub niedostateczne kompetencje zespołu, które nie pokrywają określonych aktywności, mogą wskazywać na obszar outsourcingu tych aktywności projektowych.

## 1.Wprowadzenie

Poszukiwania zmierzające do zwiększenia efektywności projektów na rynku Informatycznych i Komunikacyjnych Technologii[1] (IKT), są stymulatorem działań w wielu obszarach i kompetencji inżynierii oprogramowania w tym metodyk zarządzania projektami. Wobec rosnącej konkurencji korporacji IKT, presji czasu na kończenie nowych projektów, globalizacji tego rynku, każda organizacja musi nie tylko budować swoje kompetencje ale je efektywnie wykorzystywać, pojawiają się pytania jak to robić?. Przyjmując lub planując projekt do realizacji, kiedy oraz w oparciu o jaką informacje zdecydować, czy wszystkie jego oczekiwane produkty, pożądane aktywności wymagane do realizacji powinny być realizowane w naszej organizacji? W obszarze poszukiwania efektywnych metod wytwarzania kodu programu mamy szereg metodyk między innymi, architektura kierowana modelem (ang. *Model Driven Architecture – MDA*), jest ona podejściem do procesu wytwarzania oprogramowania dla biznesu, bazującym na automatycznych narzędziach, mającym na celu budowanie modeli niezależnych od technologii i przekształcanie ich w efektywne implementacje. Określenie „kierowana modelem" w przypadku MDA oznacza, że architektura ta udostępnia środki pozwalające użyć modeli do sterowania sposobem rozumienia, projektowania, rozwijania, wdrażania i modyfikacji systemów za jej pomocą konstruowanych. Zwiększa bardziej niż kiedykolwiek znaczenie modeli w procesie wytwarzania oprogramowania dostarcza także wytycznych w jaki sposób strukturalizować specyfikacje wyrażone w postaci modeli [1]. Z dekompozycji projektu w obszaru zarządzania całym cyklem życia projektu, wiadomo, że w zależności od charaktery projektu, sektora zastosowań wytworzenie kodu oprogramowania może stanowić znikomą część kosztów całego projektu [5]. Rozważenie wybranych aspektów projektów IKT, w obszarze biznesowym, które prowadzą do kosztów, jakie będzie ponosić organizacja przy jego realizacji oraz do działania, których rezultatem ma być wykonanie projektu przy założonym budżecie na czas oraz oczekiwanym zakresie, jest przedmiotem niniejszego artykułu.

## 2. Monitorowanie kosztów projektów

Poszukiwanie skutecznych metod i narzędzi do szacowania oraz monitorowania kosztów wszystkich projektów w firmie jest czynnikiem konkurencji, które są monitorowane na różnym etapie realizacji tj (*inicjowanie* - projekty $X_1$, $X_2 \ldots X_m$, *realizacja* - projekty $P_1$, $P_2 \ldots P_n$, *zamykanie* - projekty $Z_1$, $Z_2 \ldots Z_k$). Firma chce również wiedzieć, realizacja których projektów jest dla niej korzystna (przynosi zyski) a do których dokłada (przynoszą straty). Jakie projekty (klasa, specyfika, branża itd.) leżą w zakresie *core business* firmy, czyli są realizowane

[1] Jest to polski odpowiednik terminu szeroko używanego w literaturze anglojęzycznej: ICT - information and communication technologies

lepiej i taniej od konkurencji, a np. które komponenty projektów powinna „outsoursować”? Odpowiedź na takie pytania pojawiają się w trakcie zarządzania nie tylko jednym projektem, lecz przede wszystkim wieloma projektami. Wiążą się z tym szczególnie zagadnienia:

- Problematyki organizacji pracy nad projektami, zużywania zasobów oraz kosztów tych projektów
- Monitorowania wartości uzyskanej
- Monitorowania kosztów pracy i zużywania zasobów
- Rachunek kosztów działań w projektach informatycznych.
- Przydział kosztów stałych i kosztów pośrednich firmy do poszczególnych projektów.
- Określania opłacalności realizowania poszczególnych projektów przez te samą firmę.
- Narzędzia wspomagającego pracę PM (ang. Project Managera) w szacowaniu kosztów projektów prowadzonych w jednej firmie.

Powyższe tezy na ogół stanowią podstawę analizy i wyboru narzędzi oraz metodyki do wspomagania rozwiązywania w/w problemów.

## 3. Rachunek kosztów działań w projektach informatycznych

Metoda ABC (ang. Activity Based Costing) – rachunku kosztów działań, została uznana za jedną z największych innowacji w zarządzaniu w ostatniej dekadzie XX wieku [8].
W przeciwieństwie do tradycyjnych metod szacowania kosztów ABC pozwala spojrzeć na koszty z innej perspektywy – z perspektywy aktywności, które związane są w wytwarzaniem, przez co sprawia, że oszacowane wartości działań w firmie stają się bardziej precyzyjne.

Wdrażanie metody ABC w firmach oraz zintegrowanie jej z systemem sprawozdawczym [9,10] jest czynnością wymagającą pracy specjalistów oraz działań iteracyjnych. Wyszczególniając aktywności projektowe w systemie ABC jest jednym z proponowanych sposobów wykorzystania metody ABC w projektach informatycznych [13,14].


**Rysunek 1.** Metoda ABC w analizie kosztów projektów [5].

Podjęcie się wykonania projektu przez firmę, nawet jeśli w całości projekt leży w spektrum jej tzw. *core biznesu*, musi zostać uzasadnione przez analizę kosztów związanych z zasobami oraz czynnościami koniecznymi do realizacji projektu.

Analiza metodą ABC daje jasną odpowiedź na pytanie: „Czy koszt realizacji projektu, przy zaangażowaniu wymaganych zasobów, czasu, dodatkowych szkoleń, przerw i zaburzeń w dotychczasowych projektach jest współmierny do ewentualnych korzyści?"

Rys. 1 przedstawia diagram metody ABC, która dostarcza PM użyteczną informację wymaganą w określeniu udziału poszczególnych projektów o znanym ogólnym przychodzie. Analiza ABC pozwala również wdrażać rozwiązania maksymalizujące przychód dla poszczególnych projektów. ABC uwypukla zintegrowanie kosztów z zasobami, które projekt angażuje, co uwidacznia jego udział w zyskach względem pozostałych prowadzonych projektów. 20% klientów generuje zyski firmy, pozostałe 60% to klienci, których obsługa nie przynosi ani strat, ani zysków, a ostatnie 20% sprawia więcej problemów niż korzyści wynikających z ich obsługi. Metoda ABC pomaga w wydzieleniu tych grup tak, aby analiza opłacalności realizacji projektu była jednoznaczna.

Aby określić ile rzeczywiście kosztuje realizacja wybranych projektów, należy najpierw zidentyfikować wszystkie czynności i zasoby związane z każdym z nich. Zbiory tych czynności i angażowanych zasobów jako selekcje (często współdzielonych) wszystkich dostępnych zasobów/czynności, powinny być przeanalizowane pod kątem stopnia zaangażowania w poszczególnych projektach. Niezmiernie ważnym powodem użycia metody ABC jest odseparowanie zasobów i aktywności pomiędzy poszczególne projekty. Następnie należy rozpatrzyć udział poszczególnych projektów w kosztach i przychodach względem wszystkich realizowanych projektów. Jeśli projekt potraktujemy jako zbiór podprojektów a dokładniej wydzielimy aktywności, to umożliwi analizę kosztów tych aktywności z tymi, które można „kupić" na zewnątrz.

### 3.1 Dekompozycja projektu – wydzielenie aktywności projektowych

Metoda ABC zastosowana w obrębie projektu, ma na celu analizę kosztów realizacji poszczególnych produktów przedsięwzięcia. Wyodrębnienie aktywności i zasobów oraz zależności między nimi, jest realizowane przez dekompozycję projektu przy pomocy produktowego diagramu WBS (ang. Work Breakdown Strukture). Produkty, do wytworzenia których znany jest poziom i struktura angażowanych zasobów, mogą być traktowane jako komponenty.

Zastosowanie dekompozycji funkcji projektu pozwala na hermetyczne odseparowanie zespołu czynności, które mogą być w całościowo wykonane jako autonomiczny podprojekt. Możliwe jest zatem przydzielenie do realizacji odrębnego zespołu, wykonanie komponentu w innych ramach czasowych czy przekazanie do realizacji całkowicie zewnętrznemu podmiotowi. Hermetyzacja jest tym czynnikiem, który zapewnia niskie ryzyko niezgodności rozwiązań z komponentami współpracującymi ze sobą w ramach całego projektu. To obniża koszty synchronizacji poszczególnych rozwiązań, realizowanych w różnych technologiach dostępnych podwykonawcom.

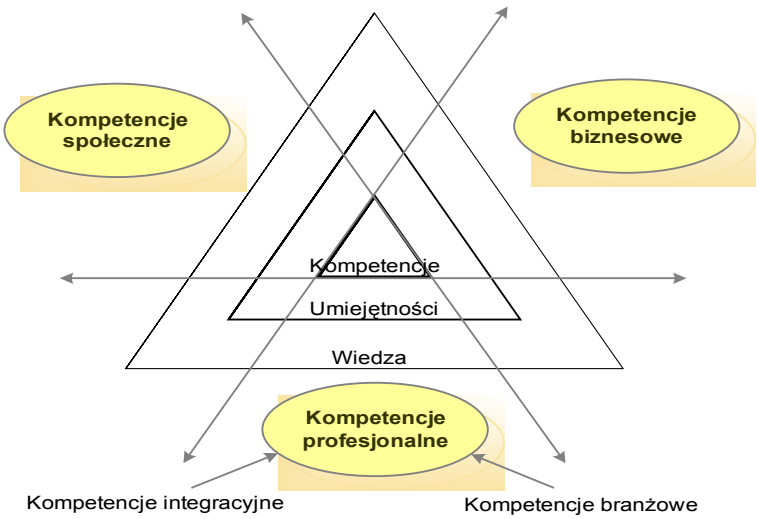
Dekompozycja projektu zmniejsza jego złożoność. Skupienie wysiłku na wytworzeniu komponentu jest łatwiejsze niż ogarnięcie całego procesu realizacji wszystkich funkcji składowych projektu.

### 3.2 Kompetencje zespołu jako jedno z ogniw decyzyjnych

Prowadzenie projektów informatycznych w nowoczesnych firmach związanych z rynkiem IT, opiera się na zespołach ludzi o zróżnicowanych kwalifikacjach. Spektrum problemów, z jakimi zespół często musi się zmierzyć, wymaga by kompetencje pracowników nawzajem się uzupełniały. Ale droga do budowy modelu zarządzania kompetencjami ich osiągnięcia jest mozolna i długa. Koncepcja zarządzania kompetencjami w obecnej postaci powstała na początku lat dziewięćdziesiątych. Pierwsze modele kompetencji zawodowych zostały opracowane w odpowiedzi na potrzeby dużych, ponad narodowych korporacji w dziedzinach związanych z nowoczesnymi technologiami. W wielu firmach menedżerowie uznali, że skala działania, zmiany technologiczne, wielokulturowość są istotnym wyzwaniem, któremu firma musi sprostać. Pierwsze prace szły w kierunku poszukiwania wzorca (modelu) menedżera, a później pracownika, tak aby zapewnić spójność kultury organizacyjnej i zarządzania w sytuacji coraz większych wyzwań.


**Rysunek 2.** Model budowy kompetencji.

Każdy członek organizacji, aby osiągnąć jedne z istotnych kompetencji dla organizacji w której pracuje przedstawione na rys. 3, według przedstawionego modelu budowy kompetencji przedstawionym na rys 2, zaczyna od zdobywania wiedzy w procesie edukacji. Wiedzę zdobywa się od czasu wynalezienia pisma a potem druku przez Jana Guttenberga nie tylko z przekazu ustnego ale z książek, szeroko rozumianej literatury oraz ostatnio Internetu. Zastosowanie wiedzy do praktycznych działań, wykonywania pracy oraz działalności twórczej dale możliwość zdobycia umiejętności w określonym obszarze wiedzy skonfrontowanej z praktycznymi działaniami na ogół pod nadzorem kogoś bardziej doświadczonego. Wreszcie dochodzimy do momentu, w którym możemy samodzielnie podejmować złożone działania przy realizacji np. projektu czyli nabywamy kompetencje.




**Rysunek 3.** Schemat ogólny klasyfikacji kompetencji [6].

Kompetencje – zdolności praktycznego wykorzystania umiejętności i wiedzy w pełni wystarczające do samodzielnego wykonywania określonego zadania w projekcie.

Przedstawiony na rys.3 przedstawiono podział kompetencji oraz wskazano na zawężanie się pól oraz zawieranie się w wiedzy umiejętności a nich kompetencji. Dla PM szczególne istotne stają się kompetencje profesjonalne w obszarze produktów branżowych oraz integracyjnych.

W przedsięwzięciach integracyjnych nie bez znaczenia są postawy prezentowane przez partnerów, osobowość PM oraz szersza wiedza wykraczająca poza wiedzę dotyczącą produktu z uwagi na konieczność prowadzenia wielu rozmów i uzgodnień nie tylko biznesowych, ale w kontekście otoczenia projektu. Uwarunkowania oraz umiejętności społecznej komunikacji i współdzielenia zainteresowań partnerów przedsięwzięcia jest kluczem zawiązania relacji, zaufania jak również współdziałania.

Kompetencje w projektach mogą być dzielone np. na trzy klasy: profesjonalne, biznesowe, społeczne.

1. Wykaz kompetencji pożądanych dla realizacji danego projektu powinien znajdować się w opisie zasobów projektu.
2. Kompetencje podlegają ciągłej ocenie oraz są wyjściem do podejmowanych przez pracownika i w stosunku do pracownika działań i decyzji.

3. Działania i decyzje mogą być powiązane z pracami w projektach u klienta lub pracami wewnętrznymi (projekty wewnętrzne budowy kompetencji).
4. Wykorzystanie kompetencji w powyższych działaniach przekłada się na stopień realizacji celów przydzielonego stanowiska w projekcie.
5. Stopień realizacji celów stanowiska przekłada się na stopień realizacji celów zespołu.
6. Stopień realizacji celów zespołu przekłada się na stopień realizacji celów projekt.

W planowaniu należy uwzględnić każdy z powyższych zasobów i zdefiniować ograniczenia, jakie wnoszą do projektu. Niektóre zasoby są czasami rozmyte lub trudno definiowalne (biznesowe, społeczne, techniczne, środowiskowe, etyczne, polityczne), ale w niektórych projektach mogą mieć podstawowe znaczenie w osiągnięciu sukcesu projektu [2, 6, 11, 15].

Powodzenie projektu wynika między innymi z poziomu kompetencji całego zespołu. Problem w budowie kompetencji firmy, która realizuje projekty polega na tym, że kierownicy projektu rekrutują się najczęściej ze ścieżki rozwoju technologicznej bądź aplikacyjno-wdrożeniowej, przez co mają ukształtowane nawyki spostrzegania projektu z perspektywy własnych doświadczeń. Ma to znaczenie w procesie zarządzania projektem. Estymacje wielu parametrów składających się na całość projektu mogą opierać się wyłącznie na subiektywnej ocenie kierownika projektu. Trafność estymacji może być różna, a charakter wykonywanego projektu może wykraczać poza zakres wiedzy i doświadczeń kierownika projektu. Dobór zespołu realizacyjnego, zorientowanego na wyniki i podnoszenie własnych kwalifikacji (kompetencji), sprzyja wprowadzeniu systemu estymacji eksperckich, gdzie poszczególni członkowie zespołu uczestniczą pośrednio w procesie harmonogramowania i monitorowania projektu. Umożliwia to znaczne obniżenie ryzyka związanego z błędną, oceną parametrów projektu określonych wyłącznie przez kierownika.

### 3.3 Co to jest model kompetencji?

Umiejętność odpowiedniego działania, zachowania, jest sumą wielu składowych, które decydują o kompetencjach. Na poziom kompetencji ma wpływ środowisko wewnętrzne, w jakim dana osoba się znajduje, kultura i wartości uznawane przez firmy.

Rozwijanie i dostosowywanie modelu kompetencji do potrzeb rozwoju firmy wspomaga optymalizowanie struktury podziału pracy, gdzie charakter realizowanych zadań odpowiada umiejętnościom osób do nich przydzielonych. Sprzyja to obniżeniu ryzyka związanego z przekroczeniem czasu bądź budżetu związanego z daną czynnością w realizowanym zadaniu.

W modelu kompetencji należy trafnie określić trzy główne obszary:

1. Definiuje, **jakie** umiejętności i kompetencje są wymagane od pracownika zajmującego określone miejsce w organizacji.
2. Określa, **kiedy** pracownik powinien posiąść dane umiejętności i poziom kompetencji w określonym zakresie.
3. Określa, **w jaki sposób** pracownik powinien stosować (‘okazywać’) posiadane umiejętności i kompetencje.

Aby wartości powyższe miały miarę ilościową, muszą istnieć narzędzia oceny kompetencji.

## 4. Narzędzie oceny kompetencji

Aby móc posłużyć się modelem kompetencji, należy zdefiniować wspólną „płaszczyznę”, dzięki której w jednolity sposób będzie można zdefiniować, ocenić i porównać umiejętności wymagane od pracownika. Tym narzędziem standaryzującym płaszczyznę oceny może być macierz kompetencji.

Macierz kompetencji stanowi język pozwalający jednakowo zidentyfikować wspólne umiejętności, wiedzę i zachowanie, które są niezbędne dla oceny kwalifikacji pracownika, do wykonywania specyficznych zadań czy określania rozwoju zawodowego pracowników. Narzędzie to dostarcza również informacji o lukach w badanych kompetencjach, wskazując precyzyjnie, które z nich powinny być wzmacniane. Ocenę kompetencji można zastosować we wszystkich obszarach funkcjonowania firmy.

Zdefiniowane kluczowe kompetencje stają się bardzo użytecznym narzędziem. Dużą zaletą modelu jest możliwość jego wykorzystania w kompleksowym systemie Zarządzania Zasobami Ludzkimi (ZZL). Począwszy od zatrudniania pracownika, poprzez ocenę jego pracy i wynagradzanie, do planowania rozwoju oraz ścieżek kariery. Macierz kompetencji wraz z właściwym zestawem technicznych i zawodowych umiejętności należy

zaprojektować do wykorzystywania dla wielu różnych potrzeb. Może być używana zespołowo lub indywidualnie na kilka sposobów takich jak: profilowanie stanowiska pracy, analiza potrzeb szkoleniowych, wybór najwłaściwszych szkoleń dla pracowników, planowanie rozwoju zawodowego, rekrutacja, rozwój zespołów i zdolności indywidualnych.


**Rysunek 4.** Model kompetencji w zarządzaniu ZZL

Umiejętności techniczne i zawodowe to umiejętności uzależnione od wykonywanej pracy lub świadczonych usług (analityk, programista, projektant, administrator). Aczkolwiek niektóre z nich mogą być wspólne dla kilku obszarów w ramach różnych aspektów działania firmy. Odpowiednie umiejętności techniczne i zawodowe powinny być przedstawiane przez właściwych kierowników odpowiedzialnych za badany obszar. Język opisujący te umiejętności powinien być zrozumiały dla pracowników danego obszaru. Umiejętności biznesowe i zachowanie są podstawą wspólnych umiejętności dla wszystkich stanowisk, począwszy od sekretariatu po stanowisko wdrożeniowca. Ocena cech przywódczych zawiera się w umiejętnościach wymaganych na stanowiskach kierowniczych średniego i wyższego szczebla. Szczegóły dotyczące kompetencji przywódczych zależą od obszaru, w którym te kompetencje są wymagane.

Pomiar i zarządzanie kompetencjami zespołu, przyczynia się do trafniejszego wyznaczania ról dla poszczególnych pracowników i jest podstawowym narzędziem ZZL. Ma to bezpośrednie przełożenie na poziom ryzyka związanego z przydziałem zasobów (oraz ich produktywnością) w realizacji przydzielonych zadań. Wprowadzenie modelu kompetencji sprzyja realokacjom i łatwiejszym zmianom w organizacji firmy. Niesie to ze sobą również zagrożenia związane z zachowaniem dotychczasowych struktur przydziału zasobów do realizowanych projektów. Jeśli tych projektów jest wiele, to zmiany alokacji pracowników mogą mieć duże znaczenie dla powodzenia realizacji rozpoczętych już przedsięwzięć. Identyfikacja zasobów krytycznych, których relokacja może okazać się zbyt kosztowna, pozwala zapobiegać sytuacjom, gdzie konsekwencje reorganizacji przydziału zasobów mogą być zbyt poważne.

Zarządzanie kompetencjami wewnątrz firmy, umożliwia szybką identyfikację ognisk podwyższonego ryzyka przy realizacji poszczególnych komponentów projektu. Jasne określenie i zmierzenie poziomu kompetencji pozwala na wyodrębnienie składowych projektu, których realizacja w zakresie działalności firmy jest zbyt ryzykowna czy kosztowna. To pozwala na sprawne wydzielenie komponentów projektu, które mogą zostać poddane procesowi outsourcingu – przekazanie innemu zespołowi czy współdzielenie ryzyka z innym partnerem. Taka sytuacja może być również podstawa podjęcia skutecznych działań w kierunku rozwija i doskonali kompetencje wykonawcze posiadanego zespołów.

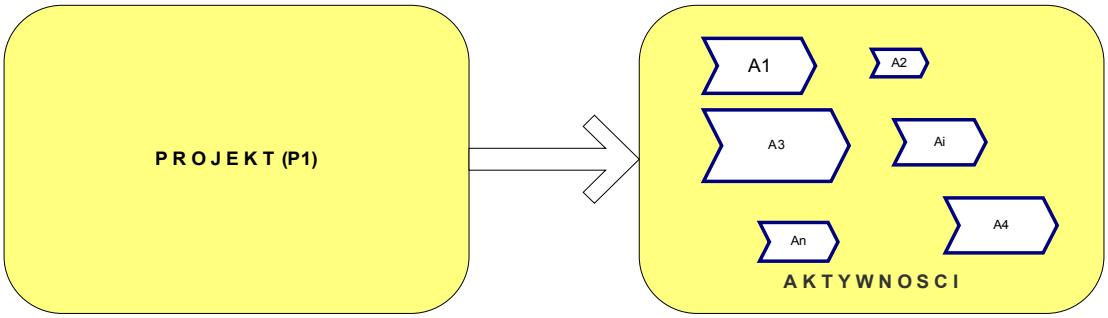
## 5. Ogólny model mapowania aktywności projektowych do posiadanych kompetencji wykonawczych

Projekty ze względu na coraz większą złożoność oraz wielkość realizowane są etapowo w kolejnych etapach cyklu życia i angażują zasoby wymagane w poszczególnych aktywnościach aby uzyskać pożądany produkt. Aktywności coraz częściej realizowane są przez liczne zespoły. Jest to możliwe poprzez dekompozycję na komponenty, do których realizacji niezbędne jest inicjowanie szeregu aktywności. Składowe projektu mogą

być różnych wymiarów, charakteryzowanych przez czynniki takie jak czas realizacji, koszt realizacji aktywności, itd. Dodatkowo można określać istotność wykonania aktywności z punktu widzenia pomyślnego ukończenia całego projektu.

### 5.1. Mapowanie aktywności projektowych

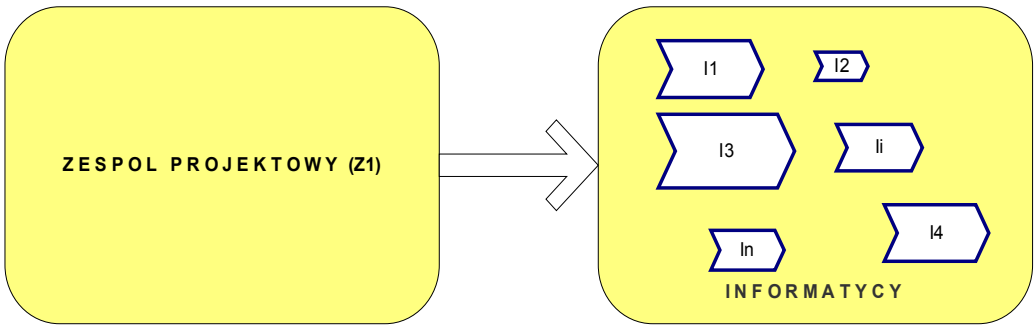
Każdy projekt P1 można podzielić na aktywności A1, A2,….An, które przy budowie planowanego komponenty mogą być różnej wielkości oraz ze względu na hermetyzacje angażują wydzielone czasowego przy ich realizacji zasoby. Lista aktywności usytuowana w przestrzeni firmy i czasie tworzy mapę aktywności projektu rys.5. Realizacja wszystkich aktywności jest konieczna do wykonania projektu.




**Rysunek 5. Dekompozycja projektu na aktywności.**

### 5.2. Mapowanie kompetencji zespołu projektowego

Organizacja, która realizuje projekty Z1 dysponuje własnymi zasobami specjalistów I1, I2,….In, których kompetencje pokrywają różne obszary aktywności biznesowych oraz profesjonalnych rys. 5. Przydzielany do projektu zespół składać się powinien z osób, które w sposób efektywny wezmą udział w realizacji projektu. Udział poszczególnych zasobów w wykonywaniu zadań projektowych (aktywności) może być różny. Członka zespołu projektowego charakteryzuje zbiór posiadanych przez niego kompetencji rozumianych jako umiejętności, wiedza, zdolności, doświadczenie, itp. Lista ulokowanych kompetencji w przestrzeni firmy i czas ich aktywnego udziału w projekcie tworzy mapę wykorzystywanych kompetencji w projekcie rys.6.




**Rysunek 6.** Dekompozycja kompetencji organizacji uaktywnianych w projekcie

### 5.3. Mapowanie aktywności z kompetencjami

**Aktywność –** wyodrębniony w ramach projektu zespół czynności, który ułatwia alokację zasobów w projekcie; komórka łatwo zarządzana, mierzalna na odpowiednim poziomie szczegółowości w oparciu o model ABC, pkt 2. Ze względu na potrzebę mierzalności atrybutów aktywności, każdą aktywność należy scharakteryzować przez łańcuch cech A1(a1, a2,….an), które utworzą macierz aktywności. Macierz aktywności składa się z pól rozmieszczonych w dwuwymiarowej przestrzeni. Poszczególne pola w zależności od położenia w macierzy oznaczają odpowiednie cechy aktywności. Cecha aktywności to właściwość aktywności niezbędna do jej

realizacji. Wartości w poszczególnych polach macierzy oznaczają ważność (wagę) cechy z punktu widzenia optymalnej realizacji danej aktywności.

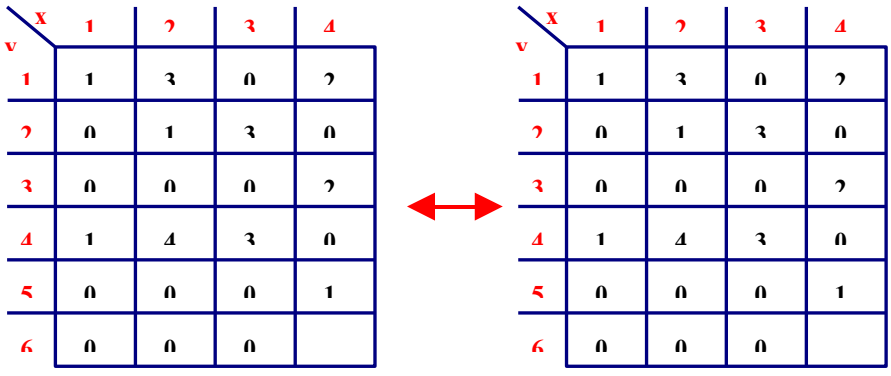
**Kompetencja** – specyficzna, zidentyfikowana, definiowalna i mierzalna wiedza, umiejętność, zdolność bądź zachowanie, którą zasób ludzki może posiadać np. w zakresie samodzielności/biegłości w danym obszarze na podstawie złożonych i realizowanych zadań oraz wsparcia udzielanego innym, która jest konieczna lub istotna z punktu widzenia wydajności realizacji konkretnego zadania w specyficznym kontekście, pkt 2.2 oraz tab.1. Ze względu na potrzebę mierzalności atrybutów kompetencji, każdą aktywność należy scharakteryzować przez łańcuch cech I1(i1, i2,….in), które utworzą macierz kompetencji. Macierz kompetencji jest zbiorem kompetencji posiadanych przez jednostkę. Konkretną kompetencję reprezentuje pole w macierzy. Wartość w polu oznacza poziom kompetencji posiadany przez opisywaną macierzą kompetencji jednostkę.

**Tabela 1**.Skala oceny poziomu samodzielności/biegłości w danym obszarze na podstawie złożoności realizowanych zadań i wsparcia udzielanego innym

| **Opis cechy wybranej kompetencji** | **ocena** |
|---|---|
| Nie mam żadnego pojęcia w tej dziedzinie | 0 |
| Mam trochę wiedzy, ale mało albo wcale praktycznego doświadczenia | 1 |
| Mam doświadczenie i mogę pracować pod warunkiem możliwości konsultacji z przełożonymi lub kolegami | 2 |
| Mam doświadczenie i samodzielnie realizuje powierzone mi zadania | 3 TAK |
| Mam szerkie doświadczenie, z którego sprawnie korzystam w złożonych sytuacjach; udzielam rad/konsultacji kolegom | 4 |
| Jestem uznanym liderem w danym obszarze; moje uwagi poszerzają zasób wiedzy dostępny w przedsiębiorstwie | 5 |

Aby sprawdzić czy i w jakim zakresie dedykowany zespół „opisany” przez macierz kompetencji jest „odpowiedni” do wykonania projektu, który jest „opisany” przez macierz aktywności, należy zbadać jakie mamy pokrycie kompetencji w stosunku do oczekiwanych aktywności poprzez proces mapowania rys 7.

Macierz mapowania jest funkcją przyporządkowania cechom aktywności odpowiednich kompetencji, dodatkowo specyfikując minimalny ich poziom. Rozważmy pole (x;y)=(1;1) w macierzy mapowania. Należy je interpretować następująco: dla cechy aktywności, oznaczonej przez pole (x;y)=(1;1) w macierzy aktywności, odpowiadająca jej kompetencja jest oznaczona poprzez pole (x;y)=(1;3) w macierzy kompetencji, przy czym poziom jej posiadania musi być co najmniej 5, aby pokrycie cechy kompetencją wynosiło 100%. Jeśli poziom jest niższy to pokrycie jest proporcjonalnie mniejsze. Suma wag wszystkich cech opisujących aktywność stanowi 100% jej pokrycia. Aby obliczyć pokrycie całej aktywności należy zsumować iloczyny pokrycia każdej cechy i procentowej wagi cechy w ramach aktywności.




**Rysunek 7**. Mapowanie aktywności i kompetencji

| y \ x | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | (1:3), 5 | (1:4), 4 | (1:1), 3 | (1:2), 5 |
| 2 | (3:2), 2 | (4:1), 3 | (5:1), 3 | (6:2), 4 |
| 3 | (5:4), 3 | (3:1), 2 | (3:3), 2 | (4:2), 3 |
| 4 | (6:1), 4 | (6:3), 5 | (2:1), 5 | (5:2), 5 |
| 5 | (2:4), 5 | (2:2), 4 | (4:3), 5 | (5:3), 5 |
| 6 | (2:3), 5 | (3:4), 5 | (4:4), 4 | ….. |

**Rysunek 8**.Macierz mapowania

Przykład:

| y \ x | 1 | 2 |
|---|---|---|
| 1 | 5 | 4 |
| 2 | 5 | 2 |

MACIERZ AKTYWNOŚCI

| y \ x | 1 | 2 |
|---|---|---|
| 1 | 2 | 5 |
| 2 | 0 | 4 |

MACIERZ KOMPETENCJI

| y \ x | 1 | |
|---|---|---|
| 1 | (1;1),3 | (2;1),5 |
| 2 | (2;2),4 | (1;2),4 |

MACIERZ MAPOWANIA

**Rysunek 8.**Badanie pokrycia kompetencjami aktywności projektowych.

**Tabela 2**.Tabela analizy pokrycia aktywności i kompetencji.

| Cechy (x;y) | Waga cechy w aktywności | Pokrycie cechy kompetencjami |
|---|---|---|
| **(1;1)** | $\frac{5}{5+4+5+2} * 100 =$ **31,25%** | Dla cechy (1;1) wymagana jest kompetencja (1;1) na poziomie, co najmniej 3. Jako, że kompetencja jest posiadana na poziomie 2, pokrycie cechy poprzez kompetencje nie będzie wynosiło 100% tylko proporcjonalnie mniej: **66,67%** |
| **(1;2)** | **25%** | **100%** |
| **(2;1)** | **31,25%** | **100%** |
| **(2;2)** | **12,5%** | **0%** |

**Pokrycie aktywności** to suma iloczynów wagi cechy w aktywności i pokrycia cechy kompetencjami dla wszystkich cech opisujących aktywność.

**Pokrycie aktywności** $= 31,25 * 66,67 + 25 * 100 + 31,25 * 100 + 12,5 * 0 = 77,08$

Jak jednak określić kompetencje opowiadające odpowiednim cechom aktywności oraz minimalny ich poziom? Początkowo tworzona jest intuicyjna macierz mapowania, w której określa się związek między cechą, a kompetencją. Poziom ustala się na najwyższy możliwy stopień posiadania kompetencji. Macierz mapowania nie jest jednak stała i jest modyfikowana w oparciu o doświadczenia pochodzące ze zrealizowanych już projektów, które są gromadzone w metrykach post-mortem (Patrz punkt 4.5. Metryki post-mortem).

### 5.4 Pokrycie projektu

Ze względu na możliwość dekompozycji projektu na aktywności, pokrycie projektu można traktować jako możliwość pokrycia wszystkich jego aktywności uwzględniając przy tym ich wagę w realizacji projektu. Wykorzystując powyższą konwencję, projekt podobnie jak aktywność członka zespołu projektowego, można opisać za pomocą macierzy. Macierz projektu jest podobna do macierzy aktywności z tym, że pole macierzy reprezentuje aktywność natomiast wartość w polu oznacza wagę aktywności z punktu widzenia realizacji projektu. Wartość zero oznacza, że aktywność nie jest wykonywana w projekcie.
W przypadku takiego opisu projektu pojawia się jedno ograniczenie: zbiór możliwych aktywności musi być skończony i ustalony z góry.

| y \ x | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 3 | 0 | 2 |
| 2 | 0 | 1 | 3 | 0 |
| 3 | 0 | 0 | 0 | 2 |
| 4 | 1 | 4 | 3 | 0 |
| 5 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | … |

MACIERZ PROJEKTU

| y \ x | 1 | 2 |
|---|---|---|
| 1 | 3 | 3 |
| 2 | 1 | 5 |

MACIERZ PROJEKTU

Przykład:

**Tabela 3**. Pokrycie aktywności w projekcie

| Aktywności (x;y) | Waga aktywności w projekcie | Pokrycie aktywności |
|---|---|---|
| (1;1) | $\frac{3}{3+3+1+5} * 100 =$ 25% | 70% |
| (1;2) | 25% | 89% |
| (2;1) | 8,33% | 95% |
| (2;2) | 41,67% | 57% |

**Pokrycie projektu** to suma iloczynów wagi aktywności w projekcie i jej pokrycia dla wszystkich aktywności wykonywanych w obrębie projektu.
**Pokrycie projektu** $= 25*70+25*89+8{,}33*95+41{,}67*57=71{,}42$

Należy rozumieć współczynnik pokrycia , że posiadamy zasoby, których kompetencje dają nam możliwość realizacji wymaganych aktywności projektu na poziomie 71,42%.

### 5.5.Metryki post-mortem

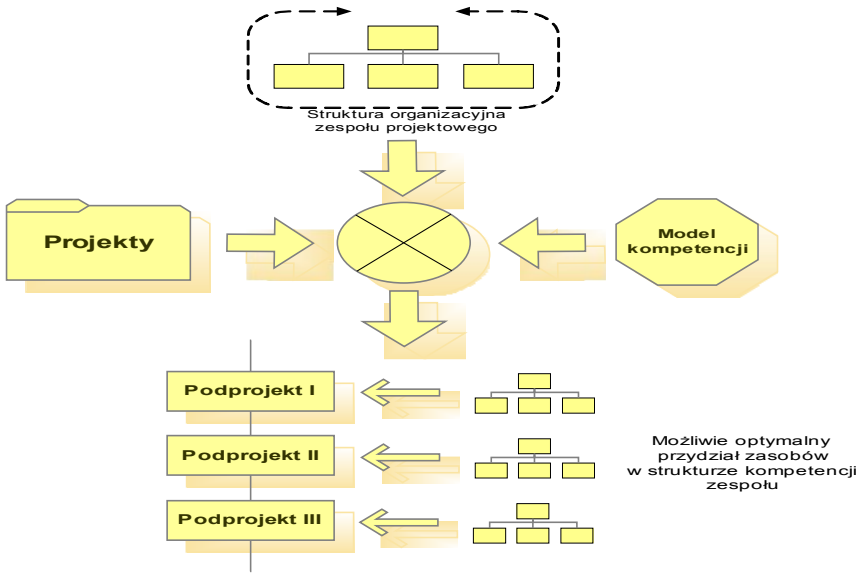
Zastosowanie metryk post-mortem w opisywanym podejściu do zarządzania projektem ma dwa zastosowania. Po pierwsze mogą być one wykorzystywane do kształtowanie macierzy kompetencji, poprzez ustalanie minimalnego poziomu posiadania kompetencji, a także modyfikowanie relacji cecha – kompetencja. Drugie zastosowanie metryk to wykorzystanie ich przy szacowaniu kosztów projektu.

## 6. Analiza ryzyka poszczególnych aktywności projektowych

Analiza pokrycia w poszczególnych aktywnościach (wydzielonych komponentów projektu), pozwala na łatwiejsze podjęcie ostatecznej decyzji odnośnie zastosowania outsourcingu. Stworzenie jednolitych metryk, umożliwiających ocenę ryzyka związanego z kompetencjami przydzielonego zespołu do realizacji komponentu, pozwala na porównanie kosztów delegowania prac na zewnątrz, a kosztami podniesienia poziomu kompetencji przydzielonego podzespołu.

Rys.10. przedstawia schemat procesu wspomagającego tworzenie metryk kompetencji dla wydzielonych komponentów. Danymi wejściowymi są projekty zawierające informacje dotyczące wymaganych poziomu kompetencji przy realizacji poszczególnych funkcji oraz schemat organizacyjny dostępnego zespołu projektowego.

Przy użyciu modelu kompetencji, następuje mapowanie podzespołów do poszczególnych komponentów projektów. Wynikowy przydział jest optymalny względem dostępnych zasobów. Metryki oceny ryzyka dotyczącego kompetencji dla poszczególnych komponentów wspomagają decyzję PM projektu IKT odnośnie przekazania realizacji w postaci outsourcingu.




**Rysunek 9.** Model minimalizowania ryzyka przez optymalny wykorzystaniu posiadanych zasobów

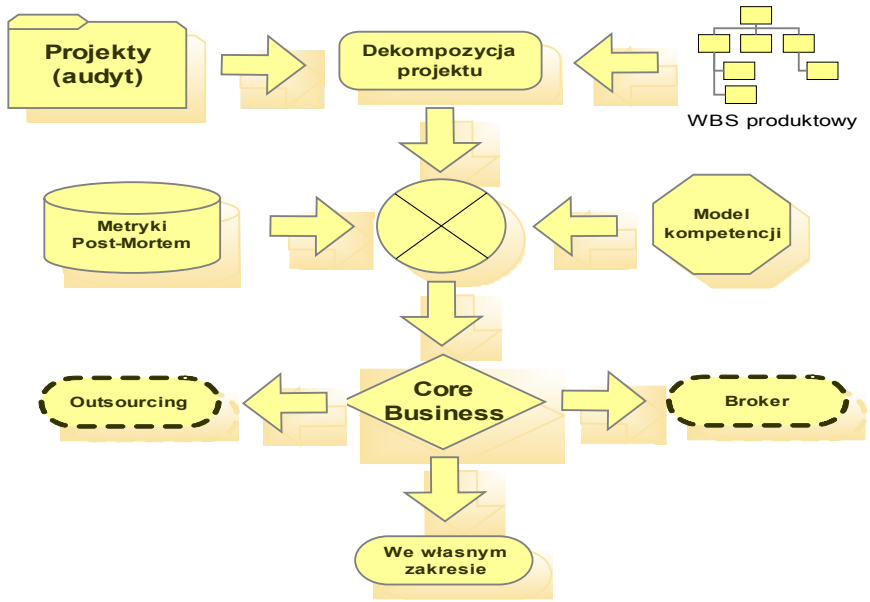
### 6.1. Outsourcing jako narzędzie minimalizacji ryzyka projektu

Istotne znaczenie w rozwoju rynku usług outsourcingowych ma wzrost świadomości informatycznej PM oraz umiejętność wyodrębnienia tych aktywności, które będą realizowane efektywniej przez wyspecjalizowane firmy zewnętrzne. Wykorzystanie outsourcingu daje możliwość skoncentrowania się na kluczowych obszarach działalności firmy. Wiąże się również ze wzrostem wydajności i efektywności procesu budowania systemów informatycznych. Olbrzymią zaletą outsourcingu jest zmniejszanie ryzyka związanego z poszczególnymi komponentami projektu poprzez jego współdzielenie z outsourcerem (odpowiedzialność materialna outsourcera za błędy i działania niezgodne z kontraktem). Outsourcing daje również możliwość łatwego dostępu do wysokiej klasy specjalistów oraz technologii, bez konieczności kosztownego podnoszenia poziomu kompetencji własnych pracowników – jest to szczególnie ważny aspekt przy realizacji projektów o wyspecjalizowanych i wąskich dziedzinach zastosowań. Trudno jest jednoznacznie wskazać miarę, na podstawie której można by ocenić potrzebę korzystania z usług outsourcingowych. Managerowie IKT muszą dokonać wnikliwej analizy nie tylko realizowanych projektów, ale również poziom kompetencji zespołu, jego zdolności do wykonania funkcji możliwych do outsourcowania i kosztów minimalizacji ryzyka z nim związanych w obrębie firmy.

Rys. 11. jest propozycją procesu wspomagającego decyzję managera IKT odnośnie outsourcowania poszczególnych funkcji projektu. Pierwszą czynnością, jaką powinien wykonać PM, jest analiza realizowanych projektów oraz ich dekompozycja do hermetycznych komponentów, które w całości mogą być poddane proce-

sowi outsourcingowania. Następnie na podstawie bazy danych metryk post-mortem dotyczących realizowanych projektów w obrębie firmy, oraz aktualnej struktury kompetencji wyłonione zostaną te komponenty, które są obarczone najwyższym ryzykiem organizacyjnym bądź technologicznym. Te komponenty są potencjalnymi kandydatami do outsourcingu.

W przypadku metryk post-mortem brane powinny być pod uwagę również aspekty związane z outsourcingiem dotychczas przeprowadzanym przez firmę w tych projektach. Jednym z istotnych kryteriów wyboru dostawcy usług outsroucingowych jest jego wiarygodność, dokładne i umiejętne wykonywanie powierzonych zadań. Występowanie tych cech jest stwierdzane na podstawie obserwacji realizacji wcześniejszych umów kooperacyjnych, wizyt i analizy list referencyjnych. Patrząc przez pryzmat *core biznesu* firmy, PM projektu IKT podejmuje decyzję o ewentualnym dodatkowym ubezpieczeniu na wypadek zajścia niepożądanych sytuacji związanych z wysokim ryzykiem wyselekcjonowanych funkcji.




**Rysunek 10**. Minimalizacja ryzyka projektów – outsourcing.

Innym rozwiązaniem jest wykonanie czynności we własnym zakresie, co wiąże się często z podniesieniem poziomu kompetencji zespołu oraz przeznaczeniem budżetu adekwatnego do pożądanego obniżenia poziomu ryzyka. PM może również przekazać wybrane komponenty do firm zewnętrznych, współdzieląc ryzyko z firmą outsourcingową.

## Podsumowanie

Włączenie się do czołówki sektora IKT jest wyzwanie dla wielu firm i korporacji tego rynku oraz państw, ponieważ w 1999 r. 17 krajów osiągnęło już 82% udziału w światowym eksporcie urządzeń informacyjnych (*IT-goods*).[2] Sektor usług IKT i sprzętu informacyjnego jest wysoce umiędzynarodowiony, toteż duża część produkcji i sprzedaży realizowana jest przez zagraniczne filie korporacji na obcych rynkach, widzimy taką działalność również w Polsce. Ponieważ w dłuższej perspektywie same mniejsze koszty pracy (USA, Angla, Niemcy a Polska, Ukraina, Indie, itp) z uwagi na zjawiska globalizacyjne nie wystarczą, dlatego poszukiwanie i wdrażanie dobrych praktyk w zarządzaniu projektami jest istotne na naszym rynku IKT.

Instrumentami zmniejszającymi ryzyko przekroczenia budżetu czy czasu w projekcie oraz dostarczenie go klientowi zgodnie z oczekiwanym zakresem i jakością, to przedmiot coraz szerszego zainteresowania kierowników projektów informatycznych. Zwiększenie bezpieczeństwa realizacji projektów informatycznych miedzy innymi poprzez rozwój kompetencji zespołu realizującego projekty informatyczne jest jednym z istotnych za-

[2] Były to następujące kraje: USA, Japonia, Tajwan, Niemcy, Malezja, Wielka Brytania, Singapur, Korea, Meksyk, Francja, Holandia, Filipiny, Tajlandia, Kanada, Hongkong, Węgry. http://www.outsourcingpipeline.com

gadnień mających duże znaczenie w budowaniu konkurencyjnej firmy na rynku IKT. Ponieważ wdrażanie modelu kompetencji oraz zarządzania wiedzą poprzez opracowanie ścieżek rozwoju dla pracowników jest dość kosztownym przedsięwzięciem, zatem najczęściej jest wprowadzane w dużych firmach IKT. Największą poprawę wskaźników: procent projektów zakończonych sukcesem oraz kosztu wytwarzania, odnotowano w dużych firmach. Natomiast w małych firmach zauważono dość znaczący (50%) wzrost kosztów przy niewielkiej poprawie wskaźnika projektów zakończonych sukcesem. Standish Group stwierdza, że istnieją trzy główne przyczyny poprawy kończenia projektów sukcesem i do nich należą:

1. Obserwowany trend dekomponowania projektów na mniejsze aplikacje
2. Ogólny wzrost umiejętności i kompetencji kierowników projektów (Certyfikacja na Poject Managera – PM) oraz postęp nauki w dziedzinie zarządzania projektami
3. Upowszechnianie standardów i narzędzi wspomagających zarządzania projektami w tym monitorowania kosztów

Myślę, że należy postawić wniosek, że czwartym czynnikiem, który zwiększy wydajność i efektywność projektów IKT tj. zmniejszenie kosztów, jest konieczność:

4. Zwiększenie automatyzacji procesów wytwórczych i większe wykorzystanie gotowych (z katalogu) komponentów.

Trzy pierwsze zalecenie – czynniki poprawiające statystyki projektów zakończonych sukcesem, zawdzięczamy coraz silniejszemu oddziaływaniu stowarzyszeń i organizacji skupiających PM oraz rozpowszechnianiu się standardów [12,16]. Należy oczekiwać, że dalsza konsolidacja rynku IKT oraz umocnienie się przedstawionych tendencji, zamieni marsz ku klęsce na marsz do sukcesowi [17].

## Literatura


1. Booch G. et al.: *MDA Manifesto;* MDA Journal, May 2004.
2. Chrościcki Zdzisław.: *Zarządzanie projektem – zespołami zadaniowymi.* Wyd. C.H. Beck, Warszawa 2001.
3. Chylewska J.: *Jak walczyć o najlepszych w firmie, jak zatrzymać kluczowych pracowników,* Materiały firmy. Hewitt Associates 2000: http://was.hewitt.com/hewitt/worldwide/europe/poland/articles/publikacje/jak_zatrzyac_kluczowych.pdf
4. Frączkowski K.: *Zarządzania projektem informatycznym.* Oficyna Wydawnicza Politechnika Wrocławska, 2003.
5. Frączkowski K., Kowalczyk M.: *Alokacja kosztów pośrednich w zarządzaniu projektami informatycznymi.* W: Efektywność zastosowań systemów informatycznych 2004.Red. Janusz K. Grabara, Jerzy S. Nowak. Warszawa: WNT 2004.
6. Frączkowski K., *Zarządzanie projektem informatycznym poprzez prognozowanie monitorowanie czynników sukcesu.* W. Koncepcje i narzędzia zarządzania wiedzą. Wydawnictwo Akademii Ekonomicznej , Wrocław 2004.
7. Frączkowski K., *Zarządzanie projektami informatycznymi w warunkach globalizacji rynku IT.* W VI Krajowa Konferencja Inżynierii Oprogramowania. Inżynieria oprogramowania. Nowe wyzwania. Red J. Górskiego i A. Wadzińskiego. Warszawa :WNT 2004.
8. Robert S. Kaplan, Robin Coope, *Zarządzanie kosztami i efektywnością,* Oficyna Ekonomiczna, 2002.
9. Casper Jones, *Activity Based Software Costing,* IEEE, 1996.
10. Pniewski K.: *Koszty działań pod kontrolą,* PC Kurier nr 22/2000.
11. Sajkowicz Alicja.: *Zasoby ludzkie w firmie.* Wyd. Poltex, Warszawa 2000.
12. Szyjewski Zdzisław.: *Zarządzanie Projektem Informatycznym.* Agencja Wydawnicza Placet, Warszawa 2001.
13. Humberto Villarreal, *The Application of Activity Based Management to a Software Engineering Environment,* IEEE, 1993.
14. Webster J.L, Reif W. E., Bracker J.S.: *The Manager's Guide To Strategic Planning Tools And Techniques,* Planning Review, Nov/Dec 1989 .
15. www.standishgroup.com
16. www.spmp.org.pl
17. Yourdon Edward, *Marsz ku klęsce* WNT 2000.

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 73–85 

# Ontology-based Stereotyping in a Travel Support System

Maciej Gawinecki[1], Mateusz Kruszyk[1], and Marcin Paprzycki[2]

[1] Adam Mickiewicz University, Poznan, Poland
[2] SWPS, Warsaw, Poland

**Abstract.** The aim of this paper is to address the problem of user profile initialization in a travel support system. In the system under consideration, ontologically demarcated data is stored in a central repository, while user profiles are functionalized as instances of travel object ontologies. Creation of an initial user profile is achieved through stereotyping. An example of utilization of this technique, in the case of restaurant stereotypes, is presented.

## 1 Introduction

Travel support systems are considered a paradigmatic example for utilization of software agents [1]. When such system is to utilize Internet-available data, there exist at least two ways of approaching its development. First, information can be indexed (only links to data are stored). Second, information can be actually gathered. While each approach has its advantages and disadvantages, we have decided to proceed with information gathering. In this way we are able to develop a system which locally resembles a "mini Semantic Web." The Semantic Web [2] is to consist of semantically demarcated resources available in a machine-consumable form. Unfortunately, while this vision is very appealing (and a hot research area), it is far from reaching practical usability. In particular, scarcity of ontologically demarcated data hinders its further development (a classic example of a chicken and egg problem). Therefore, to be able to experiment with semantically demarcated "travel content," we have designed our system around a central repository, where content is to be collected and stored. In particular, the proposed system consists of a *content collection subsystem*, where data for the system is gathered and semantically demarcated [3–5], a *content delivery subsystem* responsible for communicating with the user [6], and a *content management subsystem* where data is stored and where its quality is assured. The top-level view of the system is represented in Figure 1. Observe, that content is gathered from (a) *Verified Content Providers* (*VCP*), representing sources of content that are consistently available and format of which is changing rarely (e.g. website of Sofitel Hotels and Resorts) and (b) *Other Sources*, that represent the remaining available content. For more information about the system see [7, 8].

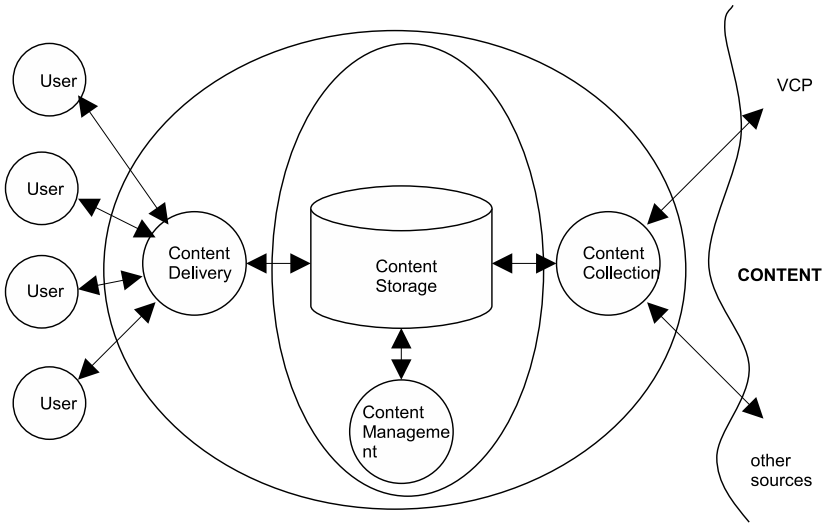



**Fig. 1.** Top level view of the system

In the context of this paper, we are particularly interested in the way that user profiles can be represented. Since data describing travel objects is collected in the form of instances of ontologies, it is natural to use the same approach to define user profiles. For instance, if one of properties of a hotel is its having a fitness center, we can specify that this feature is important to the user by giving it a high value of "liking." Therefore, user profile can be represented as an instance of a travel ontology, where user preferences are depicted as degrees of his liking (or disliking) of a given feature of a travel object.

Obviously, the aim of generating user profiles is to filter and to deliver content which is relevant to her context [9]. This can be achieved through variety of content personalization techniques. However, there it is usually assumed that (a) the system tracks user's behavior in order to learn her profile, and (b) user is willing to spend (some) time interacting with the system allowing it to "know her well." This generates a well known problem of *initialization of a new user profile* and finding solution to this problem is critical for both the user and the system [10–12]. To address it in our system we have adapted a well-known *stereotyping* approach of [13] to the Semantic Web environment that utilizes ontologically demarcated data. While our approach to user profile construction and utilization is based on ideas presented in [10, 13, 12, 14–17], utilization of these methods in the context of semantically demarcated information is novel and was proposed originally in [18].

In the paper we proceed as follows, in the next section we briefly describe ontologies that have been defined and are utilized in the system, and follow with a short discussion of user profile representation. In the next three sections we first, present some of the related work and follow with our proposed solution. In the Appendix we present more details of proposed user stereotypes.

## 2 Ontologies in the system

Our original decision, to store ontologically demarcated travel-related content, resulted in search of appropriate domain ontologies. As reported in [4, 19, 20], while there exists a large number of attempts at designing ontologies depicting various aspects of the world (including world-of travel), we were not able to locate a complete ontology of most basic travel objects, such as a *hotel* or a *restaurant*. We had therefore to define both ontologies. In the case of hotel ontology we have constructed it on the basis of information available within Internet Travel Agencies [4, 19]. For the restaurant ontology we have utilized an existing implicit restaurant ontology utilized in the ChefMoz project[3]. This ontology could not have been used directly due to the fact that data stored there is infested with bugs that make its automatic utilization impossible without pre-processing (see [4, 20] for more details). Therefore, we have reverse engineered the restaurant ontology underlying the ChefMoz project and semi-automatically cleaned data related to restaurants available in Poland. As a result, we have a fully functional and ready to experiment with dataset stored in the central repository. Below we present fragment of our restaurant ontology (in N3 notation):

```
:Restaurant   rdfs:subClassOf   loc:Location.

:accepts        a   rdf:Property;
                rdfs:label   "accepts";
                rdfs:range   mon:MeanOfPayment;
                rdfs:domain   :Restaurant.

:alcohol        a   rdf:Property;
                rdfs:label   "alcohol";
                rdfs:range   :AlcoholCode;
                rdfs:domain   :Restaurant.

:cuisine        a   rdf:Property;
                rdfs:label   "cuisine";
                rdfs:domain   :Restaurant;
                rdfs:range   :CuisineCode.

:dress          a   rdf:Property;
                rdfs:label   "acceptable dress";
                rdfs:range   :DressCode;
                rdfs:domain   :Restaurant.

:smoking        a   rdf:Property;
```

[3] *ChefMoz dining guide*, http://chefdmoz.org/.

```
rdfs:label   "smoking-friendlyness";
rdfs:range   :SmokingCode;
rdfs:domain   :Restaurant.
```

## 3 User profile representation

In order to deliver personalized content to the user we have to find a way to represent our knowledge about her, i.e. define user profile. Furthermore, its form has to simplify interactions in the system. Since our system is oriented toward processing ontologically demarcated data, it is very natural to represent user preferences in the same way. Thus we adapted an overlay model of user profile, where opinions of the user are "connected" with appropriate concepts of domain ontology. This approach is also called a *student model*, since it has been found useful to describe knowledge of the student about specific topics of the domain [21].

For instance, let us consider Ann, who does not like meat and thus she eats vegeterian food and likes salad bars. In our ontology both preferences can be expressed in terms of features of the restaurant. Specifically, we assign weight to each feature (the larger the weight the more important it is to the user). In case of Ann, vegeterian-related concepts will be assigned large weights, while other features that she does not particularly care about (e.g. disallowance of minors in a restaurant) will be assigned small weight—the lesser the interest, the closer to 0 the value will be. In the case of features for which we do not know users preferences, no value will be assigned. Let us observe that we are mimicking the notion of probability—all assigned values are from the interval (0, 1). This means that that even in the case of strong counter-preference towards a given feature we will assign it value 0 (there are no negative values available). Proceeding, we will create a special instance of restaurant ontology, one that represents user-restaurant-profile. The following "code" fragment depicts Anns profile as it would be represented in our system:

```
:AnnOpinions      a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:VeganMenu;
      sys:hasClassification sys:Interesting;
      sys:hasNormalizedProbability 0.60].
    [sys:about res:VegetarianDishes;
      sys:hasClassification sys:Interesting;
      sys:hasNormalizedProbability 0.93].
    [sys:about res:VegetarianMenu;
      sys:hasClassification sys:Interesting;
      sys:hasNormalizedProbability 0.92].
    [sys:about res:PolishCuisine ;
      sys:hasClassification sys:NotInteresting;
      sys:hasNormalizedProbability 0.11].
    [sys:about res:RussianCuisine;
      sys:hasClassification sys:NotInteresting;
      sys:hasNormalizedProbability 0.31].
    [sys:about res:MinorsNotAllowed,
      sys:hasClassification sys:NotInteresting;
      sys:hasNormalizedProbability 0.15].
    [sys:about res:SaladBar,
      sys:hasClassification sys:Interesting;
      sys:hasNormalizedProbability 0.89].
    [sys:about res:Takeout ,
      sys:hasClassification sys:Interesting;
      sys:hasNormalizedProbability 0.74].
```

Here we can find that Ann likes to eat in restaurants serving vegetarian-like food and providing take-out service; while Russian and Polish cuisines and disallowance of minors result in a "negative interest" for such restaurants.

## 4 Initialization of user profile—prior work

One of the important questions that all recommender systems have to address is: how to "introduce" new user to the system [15]. This is often called a *new user problem* and is a part of a *cold start* (or *ramp-up*) problem [12, 11]. Recommendation given by the system to the new user can be irrelevant and

result in breaking his cooperation with the system. Therefore, Montaner et al. [10] suggested using one of the following techniques to initialize user profile: manual initialization, training set or stereotyping. Our choice is stereotyping.

In everyday life a *stereotype* is a generalization about a person or group of persons. We develop and utilize stereotypes when we are unable or unwilling to obtain all information needed to make fair judgments about people or situations. In many cases, in absence of a "complete picture," stereotypes allow us to "fill in the blanks" [22]. It is one of the most common techniques used to create a model or an opinion about someone or something we do not know precisely. For example when we see a *judge* we can think about him as a well-educated, honest man and when we see a *politician* we are likely to think that he is hollow and hypocritical (although this could be incorrect). One of ways in which stereotypes help to simplify the world is that they have strong effect on what characteristics of a person or a thing are attended to and remembered [13].

In recommending systems, such as our Travel Support System, stereotyping is a method for classifying users into categories and then making predictions based on stereotypes associated with particular categories. Stereotypes contain typical assumptions that one makes about members of that category [23]. Use of stereotypes in recommending systems that maintain profiles of their users was introduced by Rich in the GRUNDY system [13], where users were assigned to one or more stereotypes and when appropriate activation condition (*trigger*) occurred, corresponding stereotype was applied. In such a stereotype-based system users can be classified either (a) on the base of answers they gave during first use of the system or (b) by tracking their behavior while they interact with the system. However, we need to remember that answering to a large number of questions can be confusing and irritating. Moreover, people asked about themselves have problems with delivering accurate information [13, 15] — sometimes people are over influenced by a group they belong to; sometimes they are afraid to confirm theirs interests etc. This fact forced us to collect information from only few explicit questions — to gain basic information about the user, and to update user profile later — according to her interactions with the system. Note here, that in most systems using stereotyping, it is the only method of information filtering (also called *demographic-filtering*). Therefore, classification is rigid: each user, once classified, cannot be reclassified, and it is difficult to specify exceptions, i.e. to respect individual characteristics that a specific user is known not to share with the group [16]. Since we use stereotyping only for initialization of the user profile, our system does not fall into the rigidity-trap and allows us to constantly adjust user profiles.

## 5 Proposed solution – stereotypes

Obviously, we use the same way of representing stereotypes as we used to represent user-profiles. The only difference is that in the case of stereotypes opinions are always "radical" and thus weight of interest in a feature are either 0 or 1. Let us illustrate this by a partial depiction of a *Conservative* stereotype (again in N3 notation) and follow by the discussion of subsequent features of that stereotype.

```
:ConservativeOpinions    a sys:OpinionsSet;
sys:containsOpinion
  [sys:about res:FullBar;
    sys:hasClassification sys:Interesting;
    sys:hasNormalizedProbability 1.0],
  [sys:about res:PolishCuisine;
    sys:hasClassification sys:Interesting;
    sys:hasNormalizedProbability 1.0],
  ...
  [sys:about res:EuropeanCuisine;
    sys:hasClassification sys:NotInteresting;
    sys:hasNormalizedProbability 0.0],
  [sys:about res:ChineseCuisine;
    sys:hasClassification sys:NotInteresting;
    sys:hasNormalizedProbability 0.0],
  [sys:about res:EclecticCuisine;
    sys:hasClassification sys:NotInteresting;
    sys:hasNormalizedProbability 0.0],
  ...
  [sys:about res:BrasserieCuisine;
    sys:hasClassification sys:NotInteresting;
    sys:hasNormalizedProbability 0.0],
  ...
```

```
[sys:about mon:BankCard;
  sys:hasClassification sys:NotInteresting;
  sys:hasNormalizedProbability 0.0],
...
[sys:about mon:DinersClubCard;
  sys:hasClassification sys:NotInteresting;
  sys:hasNormalizedProbability 0.0],
[sys:about mon:Cash;
  sys:hasClassification sys:Interesting;
  sys:hasNormalizedProbability 1.0],
[sys:about res:FastFoodCuisine;
  sys:hasClassification sys:NotInteresting;
  sys:hasNormalizedProbability 0.0],
[sys:about res:HamburgerCuisine;
...
[sys:about res:PizzaCuisine;
sys:hasClassification sys:NotInteresting;
  sys:hasNormalizedProbability 0.0].
```

*Conservative*s do not like novelty. They refuse to eat meals belonging to foreign cuisines, or fast-food. They also avoid going to restaurants that accept bank cards. They love Polish national cuisine, fatty-meaty meals, combined with large amount of alcohol.

```
:ConservativeData a sys:StereotypeProfileData;
  sys:hasDressSet
    [sys:contains sys:Elegant];
  sys:hasProfessionSet
    [sys:contains sys:Handworker, sys:UnemployedJobSeeker,
      sys:PensionerAnnuitant];
  sys:hasWealthSet
    [sys:contains sys:Rich, sys:NotRich, sys:AverageRich];
  sys:hasAgeSet
    [sys:hasLeftBound 45; sys:hasRightBound 90].
```

Typically, the *Conservative* is a mature person (age of 45 or more). He or she is usually a handworker, a pensioner or annuitant. Although the *Conservative* is not rich, he or she wears elegantly.

Let us now discuss what happens when a new user logs to the system. She will be requested to fill a short questionnaire containing questions about age, gender, income level, occupation, address (matching user features defined by the user ontology), as well as questions about her travel preferences (user data: $\hat{u}$). Personal data collected through the questionnaire will be used to match a person to one of stereotypes. More precisely, a distance measure between user-specified characteristics and these appearing in stereotypes defined in the system (stereotype data: $\hat{S}$) will be calculated to find which matches her profile the closest. To achieve this, the following formula will be used:

$$d(\hat{S},\hat{u}) = \frac{\sum_{f=1}^{k} w^f \delta^f_{\hat{S}\hat{u}} d^f_{\hat{S}\hat{u}}}{\sum_{f=1}^{k} w^f \delta^f_{\hat{S}\hat{u}}}$$

where

- $d^f_{\hat{S}\hat{u}}$ is distance measure between value of $\hat{S}$ and $\hat{u}$ at the dimension of $f$ attribute—we can use this approach because the stereotype and user data can be represented as vectors:

$$\hat{u} = \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix}, \ \hat{S} = \begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \vdots \\ \hat{S}_n \end{bmatrix}$$

- $\delta^f_{\hat{S}\hat{u}} = 0$ if there is no given value for attribute $f$, either for $\hat{S}^{(f)}$ or $\hat{u}^{(f)}$ (i.e. they have not been defined in stereotype or in user data); note, that user is not obligated to enter all personal data; otherwise $\delta^f_{\hat{S}\hat{u}} = 1$,

– $w^f$ is a significance weight of attribute $f$.

Note that since not all attributes have numerical values, the way that $d^f_{\hat{S}\hat{u}}$ is computed has to be modified accordingly. Therefore we use type classification proposed in [24]. Accordingly we distinguish the following types of $\hat{u}$ attributes:

1. *nominal*, distinguishes between categories, e.g. between professions (teacher vs. manager). Elements of this type can be equal ($\hat{u}^{(f)} = \hat{w}^{(f)}$) or not ($\hat{u}^{(f)} \neq \hat{w}^{(f)}$),
2. *ordinal*, extending nominal type by possibility of ordering values according to their range ($\hat{u}^{(f)} < \hat{w}^{(f)}$, $\hat{u}^{(f)} = \hat{w}^{(f)}$, $\hat{u}^{(f)} > \hat{w}^{(f)}$). Here, possible values of the attribute are called states, for which we assign appropriate range: $r^f_i \in [1, 2, \ldots, M^f]$ is the range of value $i$ of attribute $f$. For instance, wealth attribute can be of one of the following states: not rich ($r^f_0 = 1$), average rich ($r^f_1 = 2$), rich ($r^f_2 = 3$), very rich ($r^f_3 = 4$). Finally, $M^f$ depicts size of the domain of attribute $f$ (in the above example $M^f = 4$).
3. *interval*[4], which has values that belong to the linear scale and thus we can compute the distance measure directly ($\hat{u}^{(f)} - \hat{w}^{(f)}$). The age attribute is an example here.

In the case of stereotype data ($\hat{S}$) we must note, that each attribute carries only a "single value." For this purpose, we will establish sets of possible values for nominal and ordinal types

$$\hat{S}^{(f)} = \left\{ \hat{s}^{(f)}_1, \hat{s}^{(f)}_2, \ldots, \hat{s}^{(f)}_n \right\},$$

while for the values of interval types, we will define borders of intervals considered in the system ($\hat{S}^{(f)} = \langle y_l, y_r \rangle$).

Now we can define $d^f_{\hat{S}\hat{u}}$ more precisely, depending on the type of $f$ attribute:

– when $f$ is nominal: $d^f_{\hat{S}\hat{u}} = 0$, if $\hat{u}^{(f)} \in \hat{S}^{(f)}$; otherwise $d^f_{\hat{S}\hat{u}} = 1$;
– when $f$ is interval: $d^f_{\hat{S}\hat{u}} = 0$, if $\hat{u}^{(f)} \in \hat{S}^{(f)}$; otherwise $d^f_{\hat{S}\hat{u}} = \frac{|m - \hat{u}^{(f)}|}{max^f - min^f}$, where $m = \frac{1}{2}(y_1 + y_2)$ and $max^f$ and $min^f$ are possible maximal and minimal values of $F$, respectively.
– when $f$ is ordinal: $d^f_{\hat{S}\hat{u}} = 0$, if $\hat{u}^{(f)} \in \hat{S}^{(f)}$; otherwise $d = min_{\hat{s}^{(f)} \in \hat{S}^{(f)}} |r(\hat{s}^{(f)}) - r(\hat{u}^{(f)})|$ and we can compute: $d^f_{\hat{S}\hat{u}} = \frac{d}{M^f - 1}$.

The same process is repeated for all stereotypes to find the one that fits the best. In the next step opinions of this stereotype are moved to the profile of the new user. The complete stereotyping process has been described in figure 2.

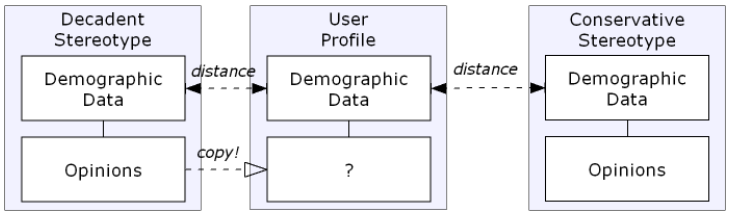



**Fig. 2.** The stereotyping algorithm.

To illustrate in more detail how the proposed method works, in Table 1 we represent Anns profile being compared with the *Conservative* stereotype.

## 6 Creating stereotypes methodological considerations

Usability and effectiveness of stereotypes depends on at least two factors: (1) quality of stereotypes themselves, and (2) quality of implications that can be drawn from the fact that someone is associated

[4] In the cited work [24] author describes also ratio type, with values, in comparison to interval type, are put on an absolute scale. Here we have joined both types into one.

**Table 1.** Example of computing distance between Ann's profile and the *Conservative* stereotype; the only similarity is found in the wealth attribute.

| *Attribute (f)* | *Conservatist's data* ($\hat{S}$) | *Ann's data* ($\hat{u}$) | *Weighted distance* ($w^f d^f_{\hat{S}\hat{u}}$) |
|---|---|---|---|
| age | 45-90 | 21 | $1.04 = 2 \cdot 0.52 = 2 \cdot \frac{\frac{1}{2}\cdot(90+45)-21}{90-0}$ |
| wealth | not rich, avarage rich or rich | rich | $4 = 4 \cdot 1$ |
| dress | elegant | natural | $0 = 1 \cdot 0$ |
| profession | handworker, unemployed / jobseeker, pensioner / annuitant; | student / pupil | $0 = 2 \cdot 0$ |
| | | *Total* ($d(\hat{S}, \hat{u})$) | $0.56 = \frac{1.04+4+0+0}{2+4+1+2}$ |

with a given stereotype [16]. The first factor is related to the number of stereotypes, correctness of selection and number of selected attributes that demarcate them, and that are used to establish distance between user profile and the stereotype. Furthermore, the selection of the significance weight ($w^f$), which specifies importance of individual attributes also plays a very important role. Psychologists Reed and Friedman [25] have shown that using normative weights to divide people into actual categories associated with their lifestyle, may result in misclassification as individuals have different self-perceptions. What is more promising is utilization of a consumer behavior model, which takes into account not only external factors, but also the self-perception that conceptualizes the way individuals think and feel about themselves, as well as how they would like to think and feel about themselves [26]. The second factor is related to the quality of available data. For instance, authors of the *LifeStyle Finder* system have utilized data from a CRM system that contained, among others, US census data, magazine subscription data, purchase data and questionnaire-based data collected for about 40,000 people [27]. In case of gastronomy, user profiles are studied only by largest restaurant chains and their data not publicly available[5]. Separately, we were not able to find non-commercial data that would allow us to connect culinary preferences with age or income.

When data is not available, it is possible to use a questionnaire. Obviously, proceeding this way will result in a substantial expenditure of time and resources [28]. The questionnaire should consist of two parts: one related to the restaurant and one to the client. In the case of our system, features of the restaurant have to match these in our restaurant ontology. As a result we should be able to answer the question: who (described demographically) is frequenting given type of restaurants. Note that the described process means also, that we will be stereotyping restaurants. In other words, to be able to build a stereotype of the user and answer the question who is coming to Japanese restaurants, we need to create a stereotype of a Japanese restaurant first [29].

Obviously, it is also possible to create stereotypes on the basis of clustering of actual user behavior. As long as a large enough data sample is available, standard clustering algorithms can be applied. Similar approach was utilized, for instance in the *Doppelgänger* system [30]. The main problem with this approach is the need to possess a large quantity of data prior to creating stereotypes. Observe however that this approach can be used to modify stereotypes in an existing system [31].

Weighting all pros and cons we noticed, that the most important drawback of stereotyping-based approaches is substantial expenditure of time and resources while constructing stereotypes. Probably, this is the reason for preparing only hand-crafted stereotypes in most of the user modeling systems [16]. However, this is usually done on the base of empirical data (e.g. purchase data), which in our case was difficult of acquire. Therefore we decided to prepare stereotypes on the base of our knowledge gathered in discussion with specialists, gastronomy literature [29] and own intuitions. We present all stereotypes in the Appendix, recognizing that each generalization could be prejudicial and misleading. However stereotyping is only the first assessment about the user in the system and other user modeling technique can improve it.

[5] information obtained from Joanna Ochniak from the School of Hotel and Restaurant Studies in Poznan, and from Andrzej Maslowski from the Institute of Market and Consumption in Warsaw; restaurant chains such as Starbucks, McDonalds or KFC have only provided us with basic information of limited practical value

## 7 Concluding remarks

In this paper we have addressed the problem of user profile initialization in a Travel Support System. We have suggested how a modified stereotyping can be used to solve this problem and presented how stereotypes can be represented and utilized in our system. Since the system is currently being implemented, in the future we will be able to report on the success of the proposed approach.

## References


1. Nwana H. S., Ndumu D. T.: *A Perspective on Software Agents Research.* The Knowledge Engineering Review **14** (1999) 1–18.
2. Berners-Lee T.: *Semantic Web* road map. `http://www.w3.org/DesignIssues/Semantic.html` (1998).
3. Pisarek S., Paprzycki M.: *Internetowy System Wspomagania Podróży—Tworzenie Podsystemu Odpowiedzialnego za Gromadzenie Danych,* In: Proceedings of the 17th Mountain Conference of the Polish Information Society. (2005) (to appear).
4. Gawineck, M., Gordon, M., Paprzycki M., Szymczak M., Vetulani Z., Wright J.: *Enabling Semantic Referencing of Selected Travel Related Resources,* In Abramowicz, W., ed.: BIS 2005: Proceedings of the 8th International Conference on Business Information Systems, Poznań, Poznań University of Economics Press (2005) 271–288.
5. Gordon M., Kowalski A., Paprzycki M., Pelech T. M, M .S., Wasowicz T.: *Ontologies in a Travel Support System,* In: Proceedings of the Internet 2004 Conference. (2005) (to appear).
6. Kaczmarek P., Paprzycki M., Gawinecki M., Vetulani Z.: *Interakcja użytkownik – agentowy system wspomagania podróży,* In: Proceedings of the 17th Mountain Conference of the Polish Information Society. (2005) (to appear).
7. Angryk R., Galant V., Gordon M., Paprzycki M.: *Travel Support System—an Agent-Based Framework.* In: Proceedings of the International Conference on Internet Computing IC2002, Las Vegas, NV, USA, CSREA Press (2002) 719–725.
8. Gordon M., Paprzycki M.: *Designing Agent Based Travel Support System,* In: Proceedings of the ISPDC 2005 Conference, Los Alamitos, CA, USA, IEEE Computer Society Press (2005) 207–214.
9. Mooers C.: *Zatocoding applied to mechanical organization of knowledge,* American Documentation (1951) 2032.
10. Montaner M., López B., de la Rosa J. L.: *A Taxonomy of Recommender Agents on the Internet,* Artif. Intell. Rev. **19** (2003) 285–330.
11. Rotaru M.: *Recommendation Systems* PhD thesis, Computer Science Department, University of Pittsburgh (2005) Comprehensive Exam.
12. Burke R.: *Hybrid Recommender Systems: Survey and Experiments,* User Modeling and User-Adapted Interaction **12** (2002) 331–370.
13. Rich E. A.: *User modeling via stereotypes,* Cognitive Science **3** (1979) 329–354.
14. Fink J., Kobsa A.: *User Modeling for Personalized City Tours,* Artif. Intell. Rev. **18** (2002) 33–74.
15. Galant V., Paprzycki M.: *Information Personalization in an Internet Based Travel Support System,* In: Proceedings of the BIS'2002 Conference, Poznan, Poland (2002) 191–202.
16. Kobsa A., Koenemann J., Pohl W.: *Personalised hypermedia presentation techniques for improving online customer relationships,* Knowl. Eng. Rev. **16** (2001) 111–155.
17. Soltysiak S., Crabtree B.: *Automatic learning of user profiles—towards the personalization of agent service,* BT Technological Journal **16** (1998) 110–117.
18. Gawinecki M., Gordon M., Paprzycki M., Vetulani Z.: *Representing Users in a Travel Support System,* In *et. al.*, H.K., ed.: Proceedings of the ISDA 2005 Conference, Los Alamitos, CA, USA (2005) 393–398.
19. Gawinecki, M., Gordon, M., Nguyen, N. T., Paprzycki, M., Szymczak, M.: *RDF Demarcated Resources in an Agent Based Travel Support System,* In: Proceedings of the 17th Mountain Conference of the Polish Information Society. (2005) 271–288 (w przygotowaniu).
20. Gordon M., Paprzycki M., Kowalski A., Pelech, T., Szymczak M., Wasowicz T.: *Ontologies in a Travel Support System,* In: Proceedings of the Internet 2004 Conference. (2005).
21. Greer J., McCalla G.: *Student modeling: the key to individualized knowledgebased instruction,* NATO ASI Series F **125** (1993).
22. Grobman G. M.: *Stereotypes and Prejudices,* `http://www.remember.org//guide//History.root.stereotypes.html` (1990).
23. McCauley C., Stitt C., Segal M.: *Stereotyping: from prejudice to prediction,* Psychological Bulletin **87** (1980) 195–208.
24. Periklis A.: *Data Clustering Techniques,* Qualifying oral examination paper, University of Toronto, Toronto, Canada (2002).
25. Reed S. K., Friedman M. P.: *Perceptual vs. conceptual categorization,* Memory & Cognition **1** (1973) 157–163.
26. Hawkins I., Best R., Coney K.: *Consumer Behavior: Building Marketing Strategy,* McGraw-Hill (1998).

27. Krulwich B.: *Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data,* Artificial Intelligence Magazine **18** (1997) 37–45.
28. Babbie E. R.: *Badania spoleczne w praktyce,* Wydawnictwo Naukowe PWN, Warszawa (2003).
29. Sala J.: *Marketing w gastronomii (wydanie I),* Polskie Wydawnictwo Ekonomiczne, Warszawa (2004).
30. Orwant J.: *Heterogeneous learning in the Doppelgänger user modeling system,* User Modeling and User-Adapted Interaction **4** (1995) 107–130.
31. Nistor C. E., Oprea R., Paprzycki M., Parakh G.: *The Role of a Psychologist in E-commerce Personalization,* In: Proceedings of the 3rd European E-COMM-LINE 2002 Conference. (2002) 227–331.

## A Stereotypes

Here we describe all stereotypes conceptualized thus far within the system with the exception of the *Conservative* stereotype, which was already described above.

**Youngster** stereotype represents people of age below 15, whose wealth can vary (since it depends on the richness of their parents). No dress style for this stereotype is specified. *Youngster*s like hamburgers, hot-dogs and pizza (fast-food). Their favorite restaurant must offer menu dedicated to kids, serve meals outdoor and be able to handle large groups. *Youngster*s usually pay by cash.

```
:YoungsterData a sys:StereotypeProfileData;
  sys:hasProfessionSet [sys:contains sys:StudentPupil, sys:UnemployedJobSeeker];
  sys:hasWealthSet [sys:contains sys:Rich, sys:NotRich, sys:AverageRich];
  sys:hasAgeSet [sys:hasLeftBound 0; sys:hasRightBound 15].

:YoungsterOpinions a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:FastFoodCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:HamburgerCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:HotDogsCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:PizzaCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about mon:Cash;
      sys:hasClassification sys:Interesting],
    [sys:about res:KidFriendly;
      sys:hasClassification sys:Interesting],
    [sys:about res:Kids;
      sys:hasClassification sys:Interesting],
    [sys:about res:Kiosk;
      sys:hasClassification sys:Interesting],
    [sys:about res:LargeGroupsOk;
      sys:hasClassification sys:Interesting],
    [sys:about res:Outdoor;
      sys:hasClassification sys:Interesting],
    [sys:about res:Family;
      sys:hasClassification sys:Interesting].
```

**Young sportsman**. This stereotype describes a person of age between 15 and 25 years, who wears sport-style clothing and is employed as a handworker. It could also be a student or someone unemployed and who therefore cannot afford expensive food. *Sportsman* likes entertainment (places that offer adult forms of entertainment and music) and eats quickly. Similarly to a *Decadent* (s)he likes drinking alcohol (mostly beer). Finally, has no problem with places that are not minority-friendly.

```
:YoungSportmanData a sys:StereotypeProfileData;
  sys:hasDressSet [sys:contains sys:SportyDress];
  sys:hasProfessionSet [sys:contains sys:Handworker,
  sys:StudentPupil, sys:UnemployedJobSeeker];
  sys:hasWealthSet [sys:contains sys:NotRich, sys:AverageRich];
  sys:hasAgeSet [sys:hasLeftBound 15; sys:hasRightBound 25].

:YoungSportmanOpinions a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:FastFoodCuisine;
```

```
    sys:hasClassification sys:Interesting],
  [sys:about res:HamburgerCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:HotDogsCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:PizzaCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:BarPubBreweryCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:MinorsNotAllowed;
    sys:hasClassification sys:Interesting],
  [sys:about res:AdultEntertainment;
    sys:hasClassification sys:Interesting],
  [sys:about mon:Cash;
    sys:hasClassification sys:Interesting],
  [sys:about res:Brewery;
    sys:hasClassification sys:Interesting],
  [sys:about res:FullBar;
    sys:hasClassification sys:Interesting],
  [sys:about res:WineBeer;
    sys:hasClassification sys:Interesting],
  [sys:about res:BeerTasting;
    sys:hasClassification sys:Interesting],
  [sys:about res:Entertainment;
    sys:hasClassification sys:Interesting],
  [sys:about res:CafeCoffeeShopCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:CafeteriaCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:TeaHouseCuisine;
    sys:hasClassification sys:NotInteresting].
```

**Student** is a person of age between 18 and 26 years old, not rich and "wearing naturally." *Student* prefers restaurants serving variety of foods: from pizza and Mexican cuisine to Asian food. His/her favorite restaurant should offer take-out and delivery services. Each meal should involve beer. A restaurant, a club or a pub is his/her second home and it is desirable from such a place to arrange concerts, provide Internet access and organize beer-tasting events. Students usually pay by cash.

```
:StudentData a sys:StereotypeProfileData;
  sys:hasDressSet [sys:contains sys:Natural];
  sys:hasProfessionSet [sys:contains sys:StudentPupil, sys:UnemployedJobSeeker];
  sys:hasWealthSet [sys:contains sys:NotRich];
  sys:hasAgeSet [sys:hasLeftBound 18; sys:hasRightBound 26].

:StudentOpinions a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:AsianCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:ChineseCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:DoughnutsCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:PizzaCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:ItalianCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:SushiCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:TexNexCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:MexicanCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:VietnameseCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:BrasserieCuisine;
```

```
    sys:hasClassification sys:Interesting],
  [sys:about mon:Cash;
    sys:hasClassification sys:Interesting],
  [sys:about res:Delivery ;
    sys:hasClassification sys:Interesting],
  [sys:about res:Takeou ;
    sys:hasClassification sys:Interesting],
  [sys:about res:Upscale ;
    sys:hasClassification sys:Interesting],
  [sys:about res:Upscale ;
    sys:hasClassification sys:Interesting],
  [sys:about res:Brewery;
    sys:hasClassification sys:Interesting],
  [sys:about res:WineBeer;
    sys:hasClassification sys:Interesting],
  [sys:about res:BeerTasting;
    sys:hasClassification sys:Interesting],
  [sys:about res:Entertainment;
    sys:hasClassification sys:Interesting],
  [sys:about res:Outdoor;
    sys:hasClassification sys:Interesting],
  [sys:about res:BeingYourOwnBeer;
    sys:hasClassification sys:Interesting],
  [sys:about res:InternetAccess;
    sys:hasClassification sys:Interesting].
```

**Snob**. Snob, often dubbed as novo-rich has accumulated wealth extremely fast. (S)he is 25-40 years old and very well to do. He or she wears elegantly. The *Snob* is employed for instance as a business-person, director of start-up company or an advertising agent. *Snob* abhors fast-food and Polish national cuisine (we consider only Polish users at this moment). However (s)he is eclectic: tries many foreign cuisines: Greek, Italian or Thai. S(he) also likes fashionable, high-value vines. (S)he pays only by a credit or a debit card.

```
:SnobData a sys:StereotypeProfileData;
  sys:hasDressSet [sys:contains sys:Elegant];
  sys:hasProfessionSet [sys:contains sys:AdvertisingMarketingWorker,
    sys:ManagerDirector];
  sys:hasWealthSet [sys:contains sys:Rich, sys:VeryRich];
  sys:hasAgeSet [sys:hasLeftBound 25; sys:hasRightBound 40].

:SnobOpinions a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:WineBeer;
      sys:hasClassification sys:Interesting],
    [sys:about res:WineList;
      sys:hasClassification sys:Interesting],
    [sys:about res:WineTasting;
      sys:hasClassification sys:Interesting],
    [sys:about res:Winery;
      sys:hasClassification sys:Interesting],
    [sys:about res:ExtensiveWineList;
      sys:hasClassification sys:Interesting],
    [sys:about mon:BankCard;
      sys:hasClassification sys:Interesting],
    [sys:about mon:ChargeCard;
      sys:hasClassification sys:Interesting],
    [sys:about mon:CreditCard;
      sys:hasClassification sys:Interesting],
    [sys:about mon:AmericanExpressCard;
      sys:hasClassification sys:Interesting],
    [sys:about mon:MasterCardEuroCard;
      sys:hasClassification sys:Interesting],
    [sys:about mon:DinersClubCard;
      sys:hasClassification sys:Interesting],
    [sys:about res:EuropeanCuisine;
```

```
    sys:hasClassification sys:Interesting],
  [sys:about res:ChineseCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:EclecticCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:ItalianCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:GreekCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:BrasserieCuisine;
    sys:hasClassification sys:Interesting],
  [sys:about res:BeingYourOwnWine;
    sys:hasClassification sys:Interesting],
  [sys:about res:InternetAccess;
    sys:hasClassification sys:Interesting],
  [sys:about res:PrivateRoomAvailable;
    sys:hasClassification sys:Interesting],
  [sys:about res:FastFoodCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:HamburgerCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:HotDogsCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:PolishCuisine;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:KidFriendly;
    sys:hasClassification sys:NotInteresting],
  [sys:about res:Kids;
    sys:hasClassification sys:NotInteresting].
```

**Decadent** is person of age between 20 and 50 years old. In the case of the *Decadent*, financial status is of no particular importance. They can be students, academicians, or free-lancers. *Decadent*s wear naturally or elegantly. They go to restaurants serving coffee, tea and wines. Places where wine could be tasted (and over-indulged) are of special interest. Fast food is disgusting for the *Decadent*.

```
:DecadentData a sys:StereotypeProfileData;
  sys:hasDressSet [sys:contains sys:NaturalDress, sys:ElegantDress];
  sys:hasProfessionSet [sys:contains sys:StudentPupil, sys:ScientistTeacher,
  sys:SpecialistFreeLancer, sys:UnemployedJobSeeker];
  sys:hasWealthSet [sys:contains sys:NotRich, sys:AverageRich];
  sys:hasAgeSet [sys:hasLeftBound 20; sys:hasRightBound 50].

:DecadentOpinions a sys:OpinionsSet;
  sys:containsOpinion
    [sys:about res:CafeCoffeeShopCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:CafeteriaCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:TeaHouseCuisine;
      sys:hasClassification sys:Interesting],
    [sys:about res:WineBeer;
      sys:hasClassification sys:Interesting],
    [sys:about res:WineList;
      sys:hasClassification sys:Interesting],
    [sys:about res:WineTasting;
      sys:hasClassification sys:Interesting],
    [sys:about res:Winery;
      sys:hasClassification sys:Interesting],
    [sys:about res:ExtensiveWineList;
      sys:hasClassification sys:Interesting],
    [sys:about res:Entertainment;
      sys:hasClassification sys:Interesting],
    [sys:about res:FastFoodCuisine;
      sys:hasClassification sys:NotInteresting],
```

```
[sys:about res:HamburgerCuisine;
  sys:hasClassification sys:NotInteresting],
[sys:about res:HotDogsCuisine;
  sys:hasClassification sys:NotInteresting],
[sys:about res:PizzaCuisine;
  sys:hasClassification sys:NotInteresting].
```

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp. 87-94
ISBN 83-922646-0-6


# Analysing system susceptibility to faults with simulation tools

P. Gawkowski, J. Sosnowski

Institute of Computer Science, Warsaw University of Technology
ul. Nowowiejska 15/19, Warsaw 00-665, Poland
jss@ii.pw.edu.pl

**Abstract.** In the paper we present original fault simulation tools developed in our Institute. These tools are targeted at system dependability evaluation. They provide mechanisms for detailed and aggregated fault effect analysis. Based on our experience with testing various software applications we outline the most important problems and discuss a sample of simulation results.

## 1 Introduction

Digital systems are widely used in various application areas including those with high reliability, availability and safety requirements (e. g. telecommunication, medicine, aviation, industrial control systems, banking). These requirements are the most important in so called dependable systems. To increase system dependability we use in general three techniques: fault avoidance, fault masking and fault tolerance [25, 28]. The main idea of fault avoidance techniques is to prevent fault occurrence. This is achieved by design reviews and automation, part selection, screening, lowering power consumption, software rejuvenation etc. Fault masking techniques hide the faults and prevent occurrence of errors using error correction codes or passive redundancy e. g. triple modular redundancy with voting. Fault tolerance techniques detect faults, identify them and perform appropriate recovery (e. g. replacing a faulty model by a spare one).

An important practical issue is to evaluate system dependability. This can be done using various analytical models, collecting reports on system operation from the field and by simulating faults and observing system susceptibility to these faults. This last approach gains high interest in recent years. Various tools have been developed for this purpose [1, 3, 5, 7, 8, 14, 24, 26]. They are targeted for models or working systems (execution-based). Simulating faults in a system model (e. g. based on hardware description language VHDL or Verilog) assures high flexibility, however, it is cumbersome for complex systems. Hence, various fault injection techniques into the working systems are widely reported in the literature [1, 3, 4, 5, 26]. In our Institute we have developed efficient tools for injecting faults into computer systems. These tools are systematically improved and used successfully in our research as well as in didactic. They comprise some original features not encountered in other tools and are recognized by other European institutions involved in dependability problems. In particular, our tools assure better experiment controllability, capability of detailed tracing of fault effects and correlation with various application properties. This has created new possibilities in dependability analysis. In section 2 we describe our fault injectors and give some illustrative experimental results and our remarks gained due to research experience with using these tools in the context of other tools. Due to some encountered problems we have developed other specialized supplementary simulation tools described in section 3. In the conclusion we summarize our experience and outline directions of our future research in this topic.

## 2 Software implemented fault injectors

In 1998 we have developed the fault injector (FITS) operating in Windows environment on IBM PC compatible platform. It has been systematically enhanced and modified. It was also the basis for other simulators targeted for multithreaded applications (MTI) and Linux environment (LIN). Using theses fault injectors for many years in student projects as well as in research, we have gained reach experience. The latest version of FITS proved that it is the one of the most complex, flexible and easy to use SWIFI (software implemented fault injector) tool reported in the literature. It uses standard Win32 Debugging API to control the execution of the software application under test. During the so called Golden Run (GR) the execution trace as well as the execution results

are saved in a log file (GRL). Additionally, statistic information is gathered on the tested application e. g. resource usage, code size, instruction distribution. FITS, as opposed to the most of other SWIFI tools reported in the literature, works on a single IBM PC compatible computer under Win32 operating systems family (Windows NT, 2000, XP). The experiments can be done on the executable code of the application (this is important if the source code is not available). However, some source code modifications (described later) can be helpful to simplify fault effect propagation analysis.

In FITS faults are simulated by disturbing the running application. In this process an important issue is the type, location and time of fault injection. To assure better experiment controllability we admit so called *testing areas*. Such areas can be marked with the use of the predefined *magic sequence* in the source code of the application (every odd occurrence of this sequence begins new testing area while every even – closes current testing area). There can be many testing areas defined within a single execution of the application. Other possibility to mark the testing area is to define two addresses of the instructions in the application code – execution of the first one starts testing area, execution of the second one – terminates it. Here no source code modifications are needed. Testing areas are very useful. They can limit the scope of disturbances only to the most interesting parts of the analyzed application. Additionally, they allow creating some extra code, not disturbed during experiments, sometimes necessary to run the analyzed application.

Another optional modification of the application to be tested is the insertion of *user-messages*, which are captured and collected by the FITS during experiments. This mechanism provides supplementary communication between the tested application and FITS and has no impact on the application behavior as the communication channel used (Win32 Debugging API) guarantees that. Using this mechanism, the tested application can signal detected errors, the path of fault propagation, efficiency of fault forecast boundaries etc. This simplifies tracing fault effects.

An important issue in experiments with fault injectors is qualification of results. This problem was neglected in the literature due to the fact that the authors in most cases analyzed simple calculation-oriented applications where the incorrect results are easily identifiable. It is much more complicated for other classes of applications e. g. oriented at database, document or signal processing, real-time applications in process control. First of all, the result of such applications has to be identified taking into account its aspects related to value, time of delivery, the impact on performed data processing etc. In some applications the accepted levels of result deviations as well as error severity should also be defined. For example, in controlling processes of a mechanical object, spurious temporary fault control signals can be tolerated by the object inertia. To resolve this problem, in FITS the correctness oracle is specified in accordance with specific features and characteristics of the tested application. Some illustration we give in [10, 11, 26].

### 2.1 Experiment setup

FITS can emulate permanent errors as well as Single or Multiple Event Upsets (SEU and MEU) related to transient faults, which dominate in contemporary systems [9, 17, 25]. Faults can be injected into the main resources available at the machine code level of the application. A fault to be injected is defined by the type of modifying operation (e. g. XOR, SET, RESET) performed on the target *fault location* (e. g. a register) in specified (in a so called bit-mask) bit positions. The experiment is configured using convenient GUI interface (an illustrative window is given in Fig.1).

The duration of a faulty state can also be programmed as a period from one machine cycle to even permanent faulty state. In FITS it is also possible to mimic the effects of complex faulty behavior or source-code level errors (e. g. execution of additional user-defined code, delays in processing paths, coding errors).

The fault injection process is preceded by execution of the analyzed application (without faults) in the so-called Golden Run mode. In this mode FITS traces the application execution and collects various statistics. The collected information helps to profile the experiments (e. g. by showing *Activity Ratio* [10, 11, 27] of resources), in particular facilitates fault distribution over execution time and resource space [11, 12, 27]. To achieve higher efficiency, the target location is disturbed just before being used (by the application under test). Faults can be located in CPU registers, application code, stack, data memory, FPU etc. Fig. 1 shows FITS dialog window for selection of fault locations. The list on the left side contains all possible fault locations. The user can select any of them and move with the buttons to the list of selected locations (list on the right side of the dialog). Faults can be defined explicitly by the operator or generated pseudorandomly. FITS assures the experiment repetitiveness, which is useful in deeper analysis of fault effects. This is a unique property, not available in similar tools reported in the literature. Injected faults can be emulated for the explicitly defined set of faults (with specified location and triggering moments) or generated automatically with a specified profile.

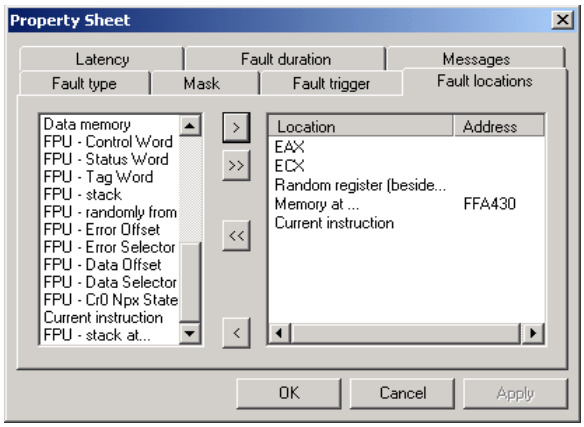



**Fig. 1.** FITS window for defining fault injections

## 2.2 Experiment reports

A fault injection experiment is composed of tests specified in some preset configuration. Each test relates to a single execution of the tested application with an injected fault (or faults). All side effects of injected faults are stored, so the fault propagation can be traced in details. However, some aggregation and filtering functions (to limit the volume of the collected data) can also be included. An example of a single fault injection test report is shown in Fig. 2.

Address of instruction at which fault was injected: 401297
Instruction execution moment: 14
Fault location: RANDOM within instruction
Instruction before first injection: 00401297: 77c9 ja 00401262
Instruction after first injection: 00401297: 37 aaa
Fault mask: randomly selected one bit
Bit disturbance operation: XOR
Fault duration: 1 instruction
Program was terminated
Exit code:254 was not correct
Messages during execution:
* Access Violation while reading - first chance at Eip=4012A3h
* Access Violation while reading - first chance at Eip=10208h
* Access Violation while reading not handled by debugee at Eip=10208h
- Second time - terminating

**Fig. 2.** A sample of a single test report

It comprises fault injection time (instruction address and its execution moment – 14th), location etc. This fault (single bit flip in instruction code) resulted in changing the instruction to be executed from conditional jump (*ja 0x00401262*) to arithmetic adjust (*aaa*). That led the application to two exceptional situations (listed as messages). The first one was generated by attempt to read unavailable memory by instruction at the address 0x004012A3. FITS gives the possibility to handle the first occurrence of exception by the application (specified as the *first chance*). Unfortunately, the injected fault generated another (second) exception - this time not handled, so at the second occurrence of the same exception the application was terminated.

For experiments with many fault injections (e. g. generated pseudorandomely in a specified resource) all tests are performed automatically and beyond detailed test reports we obtain aggregated results. In particular, we can get the percentage of results registered within specified 5 categories: C - correct result, INC – incorrect result or wrong timing of its delivery, S - fault detected by the system (specification of the number and types of registered exceptions), T - time-out, U - user-defined messages generated by the tested application (e. g. signalling faulty behaviour). Moreover, in aggregated experiment report all user-defined messages are listed

with the number of its appearance. There is also a possibility to deliver experimental results to the database for easier further processing and visualization. In embedded systems the correct result classification may admit some minor anomalies e. g. short output signal transients, delays etc. Most embedded applications run continuously but the result observation cannot be infinite process. We can limit it to some delay after the fault injection and control some internal states, which may have influence in the future operation.

### 2.3 Examples of simulation results

The usefulness of FITS was verified in many student projects and research studies. To give a better view on experimentation capabilities we give a sample of illustrative results for a calculation-oriented application. The analyzed program is the Fast Fourier's Transformation (FFT) from publicly available library – FFTW [11]. We have analyzed two versions of this program:

- V1 – the original simple implementation of FFT calculation, with neither fault detection nor fault tolerance mechanisms.
- V2 – modified application V1 comprising original fault detection and fault tolerance mechanisms introduced in [11].

Version V2 is based on time and code redundancy enhanced with exception handling. Input data is processed in iterations. Each iteration is processed twice by replicated COTS components (code and time redundancy). Moreover, it is guarded by the exception handling mechanism. This protects (with high coverage) the controlling algorithm from possible disturbances caused by a faulty processing component. The obtained pair of results is compared and in case of consensus the result is delivered as the application output. Any detected exception or disagreement between results initiates internal testing and recovery procedure (data and code consistency and recovery). It is followed by the third repetition of calculations and then voting over three results. Implementation details are given in [11].

Both versions V1 and V2 were tested with FITS and disturbed by single bit-flip faults injected randomly into the application code (instructions), CPU registers, FPU, application stack and data memory area. Around 1000 tests were made for each fault location. Experimental results are presented in Fig. 3. It shows the percentage of basic test categories (C – correct, INC – incorrect, S –system exceptions, T – time-outs) depending upon the location of injected faults (instructions, registers etc.). For each location we give a pair o bars. The left one corresponds to the V1 version and the right one to the V2 version.

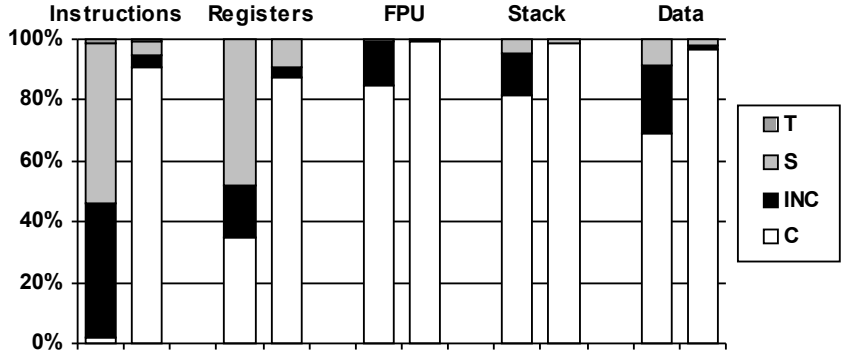



**Fig. 3.** Fault susceptibility of two versions of FFT algorithm

The experiments proved high efficiency of implemented fault tolerance mechanisms for the considered fault model. We have observed that some exceptions are not intercepted at the application level (to perform correction), so the application is terminated by the operating system. This results from some discrepancies of Windows, hence, further research targeted at improving this issue is required. Tab. 1 gives a representative distribution of not handled exceptions in Win32 environment as a consequence of single bit-flip faults in the instruction codes.

**Table 1.** Distribution of exceptions

| Exception | % |
|---|---|
| ACCESS VIOLATION | 80.6 |
| ILLEGAL INSTRUCTION | 5.5 |
| STACK OVERFLOW | 4.8 |
| BREAKPOINT | 4.7 |
| PRIVELAGE INSTRUCTION | 4.3 |
| ARRAY BOUNDS EXCEEDED | 0.1 |
| INTEGER OVERFLOW | 0.1 |

In the case of faults injected in the application code the most effective system mechanism of fault detection is Access Violation. The memory management unit triggers it, if the application violates memory access rights. Significant percentage of access violation results from the high probability of disturbing referenced addresses in such a way that unreachable memory regions are targeted. Hence, the classical control flow checking mechanisms (e. g. [2, 18, 25]) are not so effective in the Win32 applications as in embedded systems, which usually do not comprise memory protection mechanisms.

In experiments with various applications we observed different fault susceptibility in different segments or modules of these applications. Moreover, the probability of fault occurrence in a memory is a function of the used memory space. Sometimes parts of the application with high fault sensitivity are executed scarcely but occupy a large memory space. So, the distribution (in time and space) of injected faults has to be programmed carefully during the experiment setup. These issues are analyzed in [11, 12, 27].

## 3 Supplementary simulation tools

To evaluate system dependability with fault injectors we have to simulate a large number of faults in order to assure statistically significant results. Moreover, to obtain representative results it is important to assure compatibility of the test profile with the operational one [3, 15, 27]. For this purpose, we can use statistics of input data or module utilisation and select representative test scenarios to cover all possible situations. In this process some coverage measures are helpful. We can deal with functional or structural coverage of the application. Functional coverage is related to the application specifications. Structural coverage can be related to program data and control flow [27]. We use a special tool, which measures such structural features as:

- *block coverage* – coverage of bocks composed of code fragments without branching (program is composed of branch free segments of code comprising entry and exit points),
- *decision coverage* – measures the portion of decisions executed during testing,
- *c-use coverage* – counts the number of combinations of an assignment to a variable and a use of the variable in a computation that is not part of conditional expression,
- *p-use coverage* - counts the number of combinations of an assignment to a variable, a use of the variable in a conditional expression and all branches based on the value of the conditional expressions,
- *all-use coverage* - c -use or p-use,
- *du-path coverage* – counts the number of paths from a variable's definition to its use, which contains no redefinition of the variable,
- *path coverage* – coverage of all allowed sequences of statements in the program (practically not used).

SWIFI injectors are universal tools with high flexibility in the area of experiment controllability and observability. This universality may create some problems in specialised fault injection strategies related to time overhead of injections as well as the necessity of complex specification of fault triggering. Hence, for some well-defined problems it is more reasonable to use dedicated fault simulators. In particular checking fault susceptibility of complex data structures used by many applications is more efficient with specialised simulators. Such complex data structures are used in applications dealing with documents and databases etc. These structures comprise information related to the processed data (e. g. text codes, graphical objects) and various control information supporting performed operations. Moreover, some redundancy or error detection and correction features are included. In contrast to calculation-oriented applications, complex data structures show higher fault robustness. In particular a large percent of faults has no influence on the operation due to some redundancy. On the other hand, these applications perform various checking functions so many faults generate messages signalling incorrect operation. The list of these messages is quite long. We have also

analysed the fault susceptibility distribution in function of fault location within the tested documents. Some areas storing control information generated most of messages.

In the specialized simulators we can adapt fault injection locations in correlation with logical structure of the analyzed data structures. On the other hand, fault propagation effects are also controlled in a more efficient and application oriented way. Here we may have specific signalling of errors by the application (error messages), identification of abnormal states (e. g. hanging, performing partially specified functions) etc. In specialized simulators we can define test scenarios in the form of a sequence of appropriate actions with appropriate data (simulation of an operator) specific for the considered class of applications. In the case of calculation-oriented applications test scenarios can be expressed only as simple categories of appropriate data sets.

In many applications, while qualifying their test results, we can admit some disturbances as acceptable e. g. in text, graphical or sound files data disturbances may reduce the quality of the represented information still preserving the logical significance (text, image or sound are recognisable). Hence, while we evaluate the test result we have to use application dependent qualification procedures. We will illustrate this for a sample of results with randomly disturbed (bit-flip errors) files by means of special simulators. These results relate to MS Excell, LATEX, graphical and multimedia sound files.

While analyzing MS Office document fault susceptibility we have injected single bit upsets randomly within the document area (equal distribution in space). For MS Excell the injected faults generated only 12.6% of errors. An interesting thing is that different document areas have various susceptibility to faults. Some are either not used or redundant so no error was reported. Detected errors by MS Excel related to mechanisms checking document format and context of control information. Quite interesting is the distribution of document errors: 17% have not been detected by the program (INC), 46% faults generated Excel messages informing on the impossibility of loading document, 20% faults were signaled as access violation, 12% faults were not visible after opening document, however, new writing of the document generated Excel warning messages or exceptions. In 5% of cases error messages (e. g. lack of free memory) were not consistent with real faults.

LATEX files comprise text, commands, mathematical equations, comments etc. Depending upon fault rate we could use the file or not (if some important structural data is disturbed). For fault rates lower than 1/128 in 90% of cases the file was available for processing. However, legible documents (with acceptable errors) appear for fault rates at the level of 1/1000. It is interesting that disturbing mathematical formulas with fault rate up to 1/64 was acceptable.

In graphical files (e. g. JPG, BMP) an important issue is the level of image disturbance. JPG files are accessible in over 90% for fault rates below 0.005 with possibility of general recognition of the image (disturbed in about 30%). BMP files are very susceptible to faults in the area of the header (even single upsets may block access to the file). Faults in data area lower image quality. For fault rates below 0.001 practically this disturbances are not visible.

Information structure *wav* for storing and processing sound comprises a header (header length, number of sound channels, coding technique, average sample rate etc.) and sound data samples. Disturbing *wav* files with transient faults we observed that for fault rates higher than 1/40 over 90% of files were accessible but the play function in most cases was rejected. For fault rate 1/128 this function practically was available. The header area is very sensitive to faults (even single upsets are critical). Faults in data part only degrade sound quality, for fault rate 1/10 the noise is very high but sound is recognizable. We have also analyzed the impact of packet losses during transmission of *wav* files via Internet. For transmitted music listeners did not observe quality degradation (subjective evaluation) up to 3% of packet losses. Losses up to 10% were acceptable. For 30% losses sound quality was very bad, nevertheless still recognizable.

## 4 Conclusion

The developed fault simulation tools have been successfully used by many M.Sc and Ph.D students in projects, diploma thesis and our research works. The obtained results attracted the attention from other institutions active in dependability research. In particular we compared simulation results obtained with model based fault injector MODELSIM [30] developed in Grenoble TIMA3 laboratory for the car immobiliser developed in our Institute. Our execution based fault injector assures much higher speed of fault injections.

SWIFI injectors assure relatively high speed. However, they have some limitations in simulating faults at the level not accessible to the programmer. Hence, in practice it is reasonable to develop hierarchical simulation. For example characterising faulty behaviour of a module by simulating faults at a lower level (e. g. electrical or RTL model). Such characterising experiments are also reported in the literature (e. g. [13, 21, 23]). Another critical problem relates to experiment result qualification. This problem is not trivial in real time applications, where some time or result value deviations can be considered as acceptable as well as short transient pulses

(tolerated by the inertia of the controlled system). This problem is correlated with selecting representative test scenario.

Recently many fault tolerance techniques in software have been proposed (e. g. [6, 11, 16, 18, 19, 20, 22, 29]). They are especially efficient for transient faults dominating in contemporary systems [9, 17, 25]. The proposed ideas in the literature can be implemented in different ways, so their optimization is needed. In this process fault injectors are indispensable. Moreover, they are useful in dependability evaluation of existing solutions.

In our further research we concentrate on increasing fault injection effectiveness by distributing the fault injection processes in a computer network. Other improvements can be achieved by extracting system behaviour in different operation time frames and faster qualification of the experiment results (e. g. by observing selected internal state variables). We also develop a new fault injector for microcontroller systems. It is based on an original RTL level simulator [31].

**Acknowledgment.** This work was supported by KBN grant 4T11C049 25.

## References

1. Arlat, J., et al.: *Comparison of physical and software implemented fault injection technique*. IEEE Trans. on Computers, Vol. **52**, No. 8, Sept. (2003) 115-1133.
2. Benso, A., Di Carlo, S., Di Nartale, G., Prrinetto, P., Tagliaferri, L.: *Control flow checking via regular expressions*. Proc. of the 10th Asian Test Symposium (2001) 299-303.
3. Benso, A., Prinetto, P.: *Fault injection techniques and tools for embedded systems reliability evaluation*, Kluwer Academic Publishers (2003).
4. Carderilli, G.C., Kaddur, F., Leanori, A., Ottavi, M., Pontarelli, S., Velzaco, R.: *Bit flip injection in processor based architectures: a case study*. Proc. of 6th IEEE On-Line Testing Workshop (2002) 117-128.
5. Carreira, J., Madeira, H., Silva, J.G.: *Xception: a technique of the experimental evaluation of dependability in modern computers.* IEEE Trans. on Software Engineering, Vol. **24**, No. 2, February (1998) 125-136.
6. Cheynet, P., et al.: *Experimentally evaluating an automatic approach for generating safety critical software with respect to transient errors.* IEEE Transactions on Nuclear Science, Vol. **47**, No. 6, Dec. (2000) 231-236.
7. Chen, M., Tsai, T. K., Iyer, R. K.: *Fault injections and tools.* IEEE Computer, April (1997) 75-56.
8. Civera, P., et al.: *Exploiting FPGA based techniques for fault injection campaigns on VLSI circuits*. Proc. of IEEE DFT Symposium (2002) 250-258.
9. Dodd, P.E., Massengill, L.W.: *Basic mechanisms and modeling of single-event upset in digital microelectronics*. IEEE Trans. on Nuclear Science, Vol. **49**, June (2003) 583-602.
10. Gawkowski, P., Sosnowski, J.: *Dependability evaluation with fault injection experiments.* IEICE Trans. Inf. & Syst., Vol. E86-D, No.12 (2002) 2642-2649.
11. Gawkowski, P.: *Analysing and enhancing fault immunity of programs in systems with COTS elements.* Ph.D. thesis, Institute of Computer Science, Warsaw University of Technology (2005).
12. Gawkowski P., Sosnowski J., Radko B., *Analyzing the effectiveness of fault hardening procedures.* Proc. of the 11th IEEE Int'l On-Line Testing Symp. (2005) 14-19.
13. Kim, S., Somani, A.K.: *Soft Error Sensitivity characterisation for Microprocessor Dependability Enhancement Strategy.* Proc. of IEEE Int. Conference on Dependable Systems and Networks (2002) 416-428.
14. Leveugle, R.: *Fault injection in VHDL description and emulation.* Proc. of IEEE DFT Symp. (2000) 414-419.
15. Madeira, H., Some, R. R., Costa, F. D., Rennels, D.: *Experimental evaluation of a COTS system for space applications.* Proc. of IEEE Int. Conf. on Dependable Systems and Networks (2002) 325-330.
16. Nicolescu, B., Velazco, R., Reorda, M. S.: *Effectiveness and limitations of various software techniques for soft error detection, a comparative study.* Proc. of 7th IEEE Int. Testing Workshop (2001) 172-177.
17. Normand, E.: *Single Event Upset at Ground Level*, IEEE Trans. Nucl. Sci., **43**, (1996) 2742-2750.
18. Oh, N., Shrivani, P. P., McCluskey, E. J.: *Control flow checking by software signature*. IEEE Trans. on Reliability, Vol.51,No.1, March (2002) 111-122.
19. Oh, N., Mitra, S., McCluskey, E .J.: *ED⁴I Error detection by diverse data and duplicated instructions.* IEEE Trans. on Computers, Vol. **51**, No.2, Feb. (2002) 180-199.
20. Oh, N., Shrivani, P. P., McCluskey, E. J.: *Error detection by duplicated instructions in super scalar processors*. IEEE Trans. on Reliability, Vol. **51**, No.1, March (2002) 63-75.
21. Pleskacz, W., Kasprowicz, D., Oleszczak, T., Kuźmicz, W.: *CMOS standard cells characterization for defect based testing.* Proc. of IEEE DFT Symp. (2001) 384-392.
22. Rebaudengo, M., Reorda, M. S., Violante, M.: *A new software-based technique for low cost fault-tolerant application.* Proc. of Annual Reliability and Maintainability Symp. (2003) 25-28.
23. Saggese, G. P., Vetteth, A., Kalbarczyk, Z., Iyer, R.: *Microprocessor Sensitivity to Failures: Control vs Execution and Combinational vs Sequential Logic.* Proc. Of Int. Conf. on Dependable Systems and Networks (2005) 760-769.
24. Samson, J.R., Moreno, W., Falquez, F.: *A technique for automated validation of fault tolerant designs using laser fault injection.* Proc. of IEEE FTCS-28 Symp. (1998) 162-167.

25. Sosnowski, J.: *Transient fault tolerance in digital systems*. IEEE Micro February (1994) 24-35.
26. Sosnowski, J., Gawkowski, P., Lesiak, A.: *Software implemented fault inserters.* Proc. of IFAC PDS2003 Workshop, Pergamon (2003) 293-298.
27. Sosnowski, J., Gawkowski, P., Lesiak, A.: *Fault injection stress strategies in dependability analysis. Control and Cybernetics*, Vol. **33**, No. 4 (2004) 679-699.
28. Sosnowski, J.: *Testowanie i niezawodność systemów komputerowych*, EXIT, 2005.
29. Vargas, F., et al.: *Briefing a new approach to improve the EMI immunity of DSP systems.* Proc. of IEEE Asian Test Symposium (2003) 468-471
30. Velazco, R., Mochnacs, J., Peronnard, P., Calvo, O.*: Analysis of the criticality of transient bit-flip faults in a massive embedded application.* Proc. of IEEE Latin America Test Workshop (2004) 178-182.
31. Wilczyński, A., Sosnowski, J., Gawkowski, P.: *Flexible microcontroller simulation for testing purposes.* Proc. of IFAC Workshop on Programmable Devices and Systems (2004) 310-315.

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 95–103 

# Towards storing and processing of long aggregates lists in spatial data warehouses

Marcin Gorawski and Rafal Malczok

Silesian University of Technology,
Institute of Computer Science,
Akademicka 16,
44-100 Gliwice, Poland
{Marcin.Gorawski, Rafal.Malczok}@polsl.pl

**Abstract.** In this paper we present a comparison of two approaches for storing and processing of long aggregates lists in a spatial data warehouse. An aggregates list contains aggregates, calculated from the data stored in the database. Our comparative criteria are: the efficiency of retrieving the aggregates and the consumed memory. The first approach assumes using a modified Java list supported with materialization mechanism. In the second approach we utilize a table divided into pages. For this approach we present three different multi-thread page-filling algorithms used when the list is browsed. When filled with aggregates, the pages are materialized. We also present test results comparing the efficiency of the two approaches.

## 1 Introduction

Data warehouses store and process huge amounts of data. We are working in the field of spatial data warehousing. Our system (Distributed Spatial Data Warehouse—DSDW) presented in [3] is a data warehouse gathering and processing huge amounts of telemetric information generated by the telemetric system of integrated meter readings. The readings of water, gas and energy meters are sent via radio through the collection nodes to the telemetric server. A single reading sent from a meter to the server contains a timestamp, a meter identifier, and the reading values. Periodically the extraction system loads the data to the database of our warehouse. The data gathered in the database provide information about a given utility consumption. Thanks to this information we can analyze an average consumption of a given medium and appropriately control its production and distribution. When we want to analyze utility consumption we have to investigate consumption history. That is when the aggregates lists are useful. In case of electrical energy providers, meter reading, analysis, and decision making is highly time sensitive. For example, in order to take stock of energy consumption all meters should be read and the spatial data analyzed every thirty minutes. The data warehouse operation must be interactive. Query evaluation time in relational data warehouse implementations can be improved by applying proper indexing and materialization techniques. View materialization consists of first processing and then storing partial aggregates, which later allows the query evaluation cost to be minimized, performed with respect to a given load and disk space limitation [9]. In [1, 5] materialization is characterized by workload and disk space limitation. Indexes can be created on every materialized view. In order to reduce problem complexity, materialization and indexing are often applied separately. For a given space limitation the optimal indexing schema is chosen after defining the set of views to be materialized [2]. In [6] the authors proposed a set of heuristic criteria for choosing the views and indices for data warehouses. They also addressed the problem of space balancing but did not formulate any useful conclusions. [8] presents a comparative evaluation of benefits resulting from applying views materialization and data indexing in data warehouses focusing on query properties. Next, a heuristic evaluation method was proposed for a given workload and global disk space limitation.
In our current research we are trying to find the ways to improve the DSDW efficiency and scalability. After different test series (with variations of aggregation periods, numbers of telemetric objects etc.) we found that the most crucial problem is to create and manage long aggregates lists. The aggregates list is a list of meter reading values aggregated according to appropriate time windows. A time window is the amount of time in which we want to investigate the utility consumption. The aggregator is comprised of the timestamp and aggregated values (fig. 1). In the system presented in [3] aggregates lists are used in the indexing structure,

| TIMESTAMP | ZONE 1 | ZONE 2 |
|---|---|---|
| 2004-01-01 00:30:00 | 0.786 | 0.13 |
| 2004-01-01 01:00:00 | 0.97 | 0.65 |
| 2004-01-01 01:30:00 | 1.034 | 0.453 |

• • •

**Fig. 1.** Example of an aggregates list for time window of 30 minutes.

aggregation tree, that is a modification of an aR-Tree [7]. Every index node encompasses some part of the region where the meters are located and has as many aggregates lists as types of meters featured in its region. If there are several meters of the same type, the aggregates lists of the meters are merged (aggregated) into one list of the parent node (fig. 2). In the following sections we present and compare two different approaches

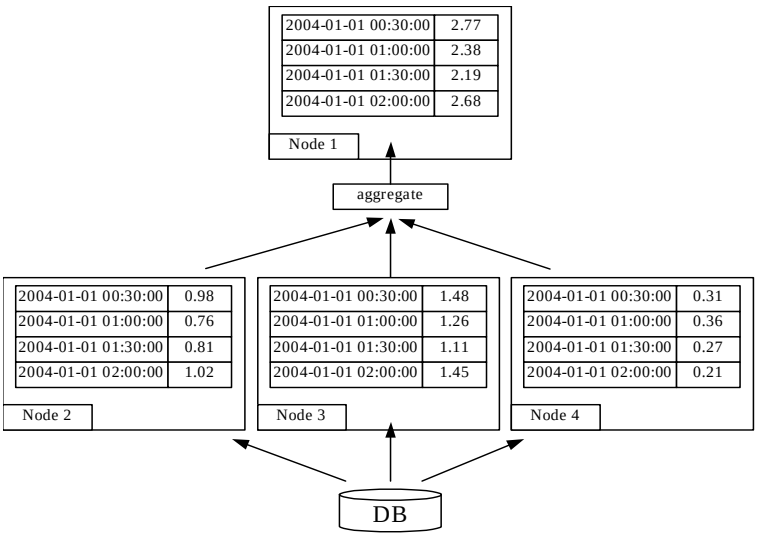



**Fig. 2.** A hypothetical indexing structure.

for storing and managing long aggregates lists. First we present the theoretical background and then we discuss the efficiency test results. Finally we conclude the paper choosing the more efficient and scalable solution.

## 2 First approach—materialized Java list

The first approach assumes using a standard Java language list (like *LinkedList* or *ArrayList*, see [10]) to store the aggregates. Created aggregates are added to the list; the list is sorted according to the timestamp. The aggregates lists are stored in the main computer memory. Memory overflow problems may occur when one wants to analyze long aggregation periods for many utilities meters. If we take into consideration the fact that the meter readings should be analyzed every thirty minutes, simple calculations reveal that the aggregates list grows very quickly with the extension of an aggregation period. For instance, for single energy meter an aggregates list for one year has $365 \cdot 48 = 17520$ elements. In order to protect the system from this failure we designed a memory management algorithm. The algorithm operation is unnoticeable for a system user.

When the system starts, the algorithm evaluates the memory capacity by creating artificially generated aggregates lists. Each allocated aggregates list is counted. The action is continued until there is no more free memory. The next step is to remove the allocated lists and compute the value indicating the maximal amount of lists in the memory during system operation. That value is computed as 50% of the all of allocated

lists. The 50% margin is set in order to leave enough memory for other system needs, e.g. user interface and insurance against memory fragmentation.
Memory overflow may occur when new aggregates lists are created. When the previously mentioned limit is exceeded (too many aggregates lists in the memory) then some aggregates lists have to be removed from the memory. In order to determine which lists to remove, we applied a mechanism that uses a lists reading counter. Every indexing structure node has the reading counter increasing each time the lists are read. The memory managing mechanism searches for the nodes with the smallest value of the reading counter and removes them from the memory. Aggregates removal is proceeded by the materialization operation. In figure 3 we present a flowchart illustrating the memory managing algorithm operation supported by the materialization mechanism.

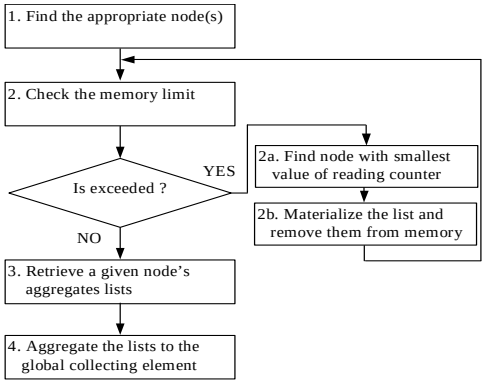



**Fig. 3.** The memory managing algorithm operation.

### 2.1 Materialization

In order to save time spend on raw data processing we decided to apply the idea of aggregates lists materialization. When a given aggregates list is created for the first time, a set of queries must be executed and the resulting raw data processed. Once calculated, the data is stored for further use. The materialization mechanism combines the Java streams and the Oracle BLOB table column. The created aggregates are stored in a table consisting of two columns, the first being NUMBER storing node's identifier and the second is BLOB storing materialized binary data of a given node's aggregates lists. Figure 4 shows a simple schema of the materialization mechanism. The presented approach of storing a single aggregates list as one stream has two main drawbacks: the list must always be read starting from the beginning what, if not necessary, can be very time-expensive, and the list cannot be easily updated (to attach new aggregators we must restore, update and store whole long list). In the next section we present a different solution to the list materialization problem.

## 3 Second approach—Materialized Aggregate List

The second approach is called a Materialized Aggregate List (MAL). Our main goal when designing the MAL was to create a list that could be used as a tool for mining data from the database as well as a component of indexing structure nodes (fig. 5). In this paper we focus mainly on the differences between the previously and currently applied solutions. For more theoretical and practical details on the Materialized Aggregate List please refer to [4]. The list interface is identical to the standard Java list so the MAL can easily replace the previously described aggregates list. We also suppose that the applied multi-thread page-filling algorithms and selective materialization will result in the MAL being more efficient when compared to the standard Java list. Our main intention when designing the MAL was to build an efficient solution free of memory

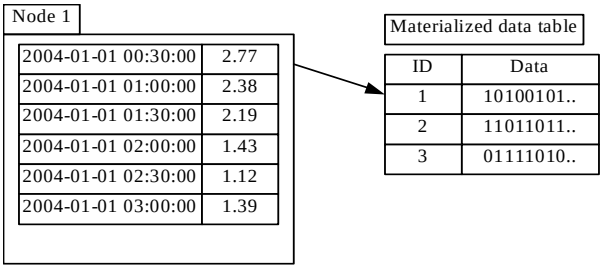



**Fig. 4.** A simple schema of the materialization mechanism.

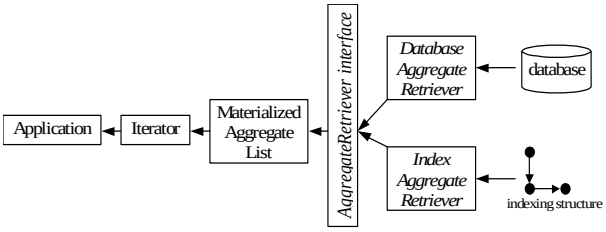



**Fig. 5.** MAL idea – provide a solution based on a well-known standard.

overflows which would allow aggregates list handling with no length limitations. The MAL structure and operation is based on the following approach: every list iterator uses a table divided into pages (the division is purely conventional). When an iterator is created some of the pages are filled with aggregators (which pages and how many is defined by the applied page-filling algorithm, see description below). Applying a multi-thread approach allows filling pages while the list is being browsed. The solution also uses an aggregates materialization mechanism to store once created aggregates.
The actual list operation begins when a new iterator is created (*iterator*() function call). Every iterator is characterized by two dates:

- border date. The border date is used for managing the materialized data. The border date is equal to the timestamp of the first aggregator in the page.
- starting index. In the case that starting date given as a parameter in the *iterator*() function call is different from the calculated border date, the iterator index is adjusted so that a the first *next*() function call returns the aggregator with the timestamp nearest to the given starting date.

Consider the following: we have the install date 2004-01-01 00:00:00, an aggregation window width of 30 minutes and page size of 240. So, as can be easily calculated, on one page we have aggregates from five days (48 aggregates from one day, $\frac{240}{48} = 5$). Next we create an iterator with the starting date 2004-01-15 13:03:45. The calculated border date will be 2004-01-11 00:00:00, starting index 218 and the first returned aggregator will have the timestamp 2004-01-15 13:30:00 and will contain the medium consumption between 13:00:00 and 13:30:00.

### 3.1 Page-filling algorithms

The iterator table pages are filled by separate threads. Regardless of the applied page-filling algorithm an individual thread, characterized by a page border date, operates according to the following steps:

1. Check whether some other thread filling a page with an identical border date is currently running. If yes, register in the set of waiting threads and wait.
2. If no, check if the required aggregates were previously calculated and materialized. If yes, restore the data and go to 4.
3. If no, fill the page with aggregates. Materialize the page.
4. Browse the set of waiting threads for threads with the specified border date. Transfer the data and notify them.

In the subsections below we present three different page-filling algorithms used for retrieving aggregates from the database. Their operation is illustrated with figures that use the symbols described in figure 6.

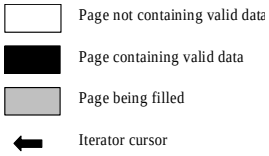



**Fig. 6.** Symbols used in the page-filling algorithms descriptions.

**Algorithm SPARE** Two first pages of the table are filled when a new iterator is being created and the SPARE algorithm is used as a page-filling algorithm. Then, during the list browsing, the algorithm checks in the *next*() function if the current page (let's mark it $n$) is exhausted. If the last aggregator from the $n$ page was retrieved, the algorithm calls the page-filling function to fill the $n+2$ page while the main thread retrieves the aggregates from the $n+1$ page. One page is always kept as a "reserve", being a spare page (fig. 7). This algorithm brings almost no overhead—only one page is filled in advance. If the page size is set appropriately so that the page-filling and page-consuming times are similar, the usage of this algorithm should result in fluent and efficient list browsing.

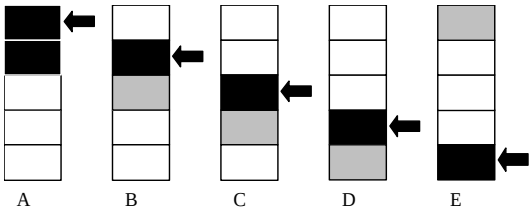



**Fig. 7.** Operation of the SPARE algorithm.

**Algorithm RENEW** When the RENEW algorithm is used, all the pages are filled during creation of the new iterator. Then, as the aggregates are retrieved from the page, the algorithm checks if the retrieved aggregator is the last from the current page (let's mark it $n$). If the condition is true, the algorithm calls the page-filling function to refill the $n$ page while the main thread explores the $n+1$ page. Each time a page is exhausted it is refilled (renewed) immediately (fig. 8). One may want to use this algorithm when the page consuming time is very short (for instance the aggregators are used only for drawing a chart) and the list browsing should be fast. On the other hand, all the pages are kept valid all the time, so there is a significant overhead; if the user wants to browse the aggregates from a short time period but the MAL is configured so that the iterators have many big pages—all the pages are filled but the user does not use all of the created aggregates.

**Algorithm TRIGG** During new iterator creation by means of the TRIGG algorithm, only the first page is filled. When during $n$ page browsing the one before last aggregator is retrieved from the page the TRIGG algorithm calls the page-filling function to fill the $n+1$ page. No pages are filled in advance. Retrieving the next to last aggregator from the $n$ page triggers filling the $n+1$ page (fig. 9). The usage of this algorithm brings no overhead. Only the necessary pages are filled. But if the page consumption time is short the list-browsing thread may be frequently stopped because the required page is not completely filled.

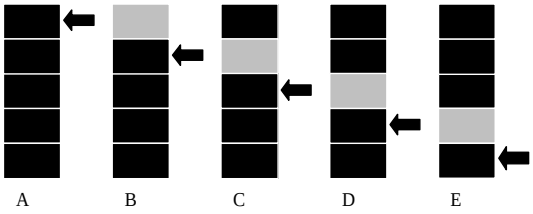



**Fig. 8.** Operation of the RENEW algorithm..

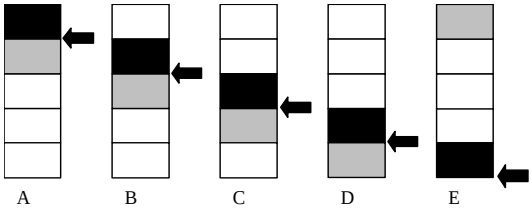



**Fig. 9.** Operation of the TRIGG algorithm.

### 3.2 MAL in indexing structure

The idea of the page-filling algorithm in the MAL running as a component of a higher-level node is the same as in the TRIGG algorithm for the database iterator. The difference is in aggregates creating because the aggregates are created using aggregates of lower-level nodes. The creation process can be divided into two phases. In the first phase the aggregates of the lower-level nodes are created. This operation consists of creating the aggregates and materialization. In the second phase, the aggregates of the higher-level node are created through merging all the available materialized pages of the lower-level nodes created in the first phase into pages of the higher-level node. The second phase is performed using the materialized data created in the first phase and its execution takes less than 10% of the whole time required for creating the aggregates of the higher-level node.
To control the number of concurrently running threads we use a resource pool storing the iterator tables. Thanks to such approach we are able to easily control the amount of memory consumed by the system (configuring the pool we decide how many tables it will contain at maximum) and the number of running threads (if no table is available in the pool a new iterator will not start its page-filling threads until some other iterator returns a table to the pool).

### 3.3 Materialization

Page-filling thread operation description shows that the MAL applies materialization. For the MAL we use a table with three columns storing the following values: the object identifier (telemetric object or indexing structure node), page border date and aggregators in binary form (fig. 10). The page materialization mechanism operates identically for each page-filling algorithm. The MAL can automatically process new data added by the extraction process. If some page was materialized but it is not complete, then the page-filling thread starts retrieving aggregates from the point where the data was not available.

## 4 Test results

This section contains a description of the tests performed for the both presented approaches. The aggregates were created for 3, 6, 9, and 12 months with a time window of 30 minutes. The created aggregates were not used in the test program; the program only sequentially browsed the list. Aggregates browsing was performed twice: during the first run the list has no access to the materialized data, and during the second run a full set of materialized data was available.The tests were executed on a machine equipped with Pentium IV 2.8 GHz and 512 MB RAM. The software environment was Windows XP Professional, Java Sun 1.5 and Oracle 9i.

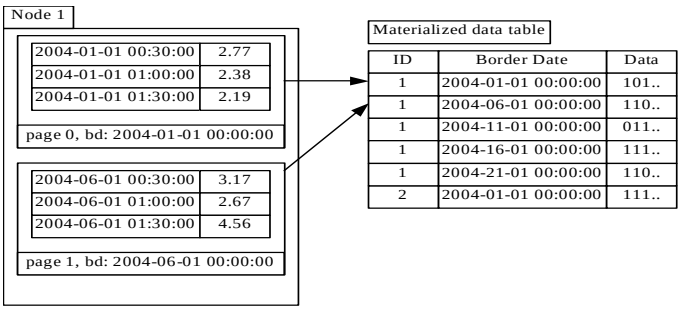



**Fig. 10.** A schema of the materialization mechanism applied in the MAL.

In the case of the first presented approach we have little influence on the list configuration. But in the case of MAL we can change the following configuration parameters to make the list operation most efficient:

- page size defines how many aggregates is stored on a single list page,
- page number defines the amount of pages creating the iterator table,
- page-filling algorithm defines which algorithm is to be used for filling the pages.

In order to compare the two presented approaches we first analyze operation of the MAL to find the best combination of the configuration parameters. The choice criterion consisted of two aspects: the efficiency measured as a time of completing the list-browsing task and memory complexity (amount of the memory consumed by the iterator table).

### 4.1 MAL configuration

The tests were performed for the three page-filling algorithms, size of a single page varied from 48 aggregates (1 day) to 4464 (93 days – 3 months) and number of pages $2 \div 10$. The number of tables available in the table pool was limited to 1 because increasing this number brought no benefit.
We first analyze the results of completing the list-browsing task during the first run (no materialized data available) focusing on the influence of the page number and the size of a single page. We investigated the relations between these parameters for all three algorithms, and we can state that in all the cases their influence is very similar; graphs of the relations are very convergent. The list browsing times for small pages are very diverse. For a page of size 48 the times vary from 30 to 160 seconds depending on the amount of pages. MAL operation for a page of size 240 is more stable; the differences resulting from the different number of pages do not exceed 25 seconds. For pages of size 672 and more we observe very stable operation of the MAL. A page size of 672 seems to be the best choice. Extending the page size brings very small efficiency gain but results in much more memory consumption. After choosing the best pair of page size and page number parameters, we compared the time efficiency of the page-filling algorithms. Figure 11 shows a graph comparing efficiency of the algorithms applied in the child nodes for browsing the list of aggregates for 3, 6, 8 and 12 months. The lists were configured to use 6 pages, each of size 672. In the graph we observe that the SPARE and the TRIGG algorithms show similar efficiency. Along with extending the aggregation period the operation time increases; for the TRIGG algorithm the increase is purely linear. The RENEW algorithm shows worse efficiency, especially for long aggregation periods of 6 and 12 months. Filling and merging the pages not used during the list browsing results in worse performance. Therefore, to summarize the parameters selection we can state that the MAL works efficiently for the following configuration: number of pages $4 \div 6$, size of a single page 672 and TRIGG as the page-filling algorithm.

### 4.2 Efficiency comparison

In this subsection we compare efficiency of the two presented approaches. The scenario concerns operation of the lists when applied in a theoretical indexing structure. The structure consisted of one parent node and $5 \div 20$ child nodes; the query concerned parent aggregates but obviously resulted in creating the aggregates

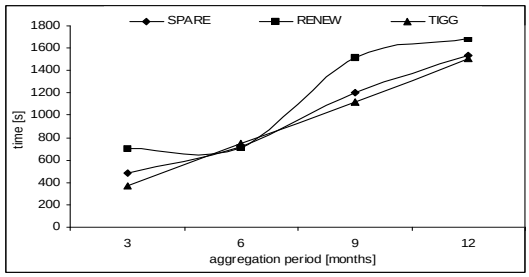



**Fig. 11.** Page-filling algorithms operation for index iterator

of all the child nodes. As first we compare the results of the lists operation during the first run when no materialized data was available. Figures 12 and 13 present graphs for gathering aggregates from respectively 5 and 20 child nodes. In the both cases the MAL shows better efficiency. It operates about 40% faster than the standard Java list supported by the materialization mechanism. Further graph analysis reveals that extending the aggregation period results in operation time growth. The growth is significantly greater in the case of the Java list. The conclusion is that the MAL solution is more scalable.

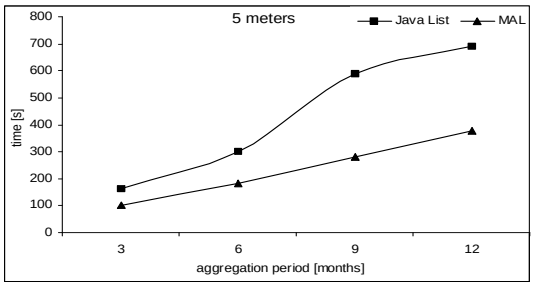



**Fig. 12.** Comparison of efficiency of the Java list and the MAL for 5 meters.

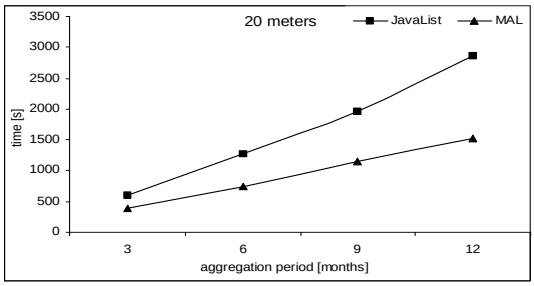



**Fig. 13.** Comparison of efficiency of the Java list and the MAL for 20 meters.

The aspect last investigated was materialization influence on system efficiency. The test results interpretation reveals that materialization strongly improves efficiency of both approaches. Thanks to materialization both lists operate from 6 to 8 times faster than when no materialized data is available. The longer the aggregation period and the more meter readings are to be aggregated the greater the benefit of materialization

## 5 Final conclusions and future plans

In this paper we presented a comparative study of two approaches for storing long aggregates lists in spatial data warehouse. The first approach is a standard Java list supported by aggregates materialization mechanism. The second solution is called Materialized Aggregate List (MAL) and is a data structure for storing long aggregates lists. The MAL also uses the materialization mechanism. After presenting the theoretical background for both approaches we discussed test results proving the MAL being more efficient and scalable. The MAL operates about 40% faster than a standard Java list.
The data warehouse structure described in [3] applies distributed processing. We suppose that in this aspect introducing the MAL to our system will bring benefits in efficiency. The current approach to sending complete aggregates lists as a partial result from a server to a client results in high, single client module load. When we divide the server response into MAL pages, the data transfer and the overall system operation will presumably be more fluent. Implementation and testing of those theoretical assumptions are our future plans.

## References

1. Baralis E., Paraboschi S., Teniente E.: *Materialized view selection in multidimensional database,* In Proc. $23^{th}$ VLDB, pages 156–165, Athens, 1997.
2. Golfarelli M., Rizzi S., Saltarelli E.: *Index selection for data warehousing,* In. Proc. DMDW, Toronto, 2002.
3. Gorawski, M., Malczok, R.: *Materialized aR-Tree in Distributed Spatial Data Warehouse,* SSDA_ECML/PKDD Workshop, Pisa 2004.
4. Gorawski, M., Malczok, R.: *On Efficient Storing and Processing of Long Aggregate Lists,* DaWaK, Copenhagen 2005.
5. Gupta H.: *Selection of views to materialize in a data warehouse,* In. Proc. ICDT, pages 98–112, 1997.
6. Labio W. J., Quass D., Adelberg B.: *Physical database design for data warehouses,* In. Proc. ICDE, pages 277–288, 1997.
7. Papadias D., Kalnis P., Zhang J., Tao Y.: *Effcient OLAP Operations in Spatial Data Warehouses,* Spinger Verlag, LNCS 2001
8. Rizzi S., Saltarelli E.: *View Materialization vs. Indexing: Balancing Space Constraints in Data Warehouse Design,* CAISE, Austria 2003.
9. Theodoratos D., Bouzehoub M.: *A general framework for the view selection problem for data warehouse design and evolution,* In. Proc. DOLAP, McLean, 2000.
10. *Java$^{TM}$ 2 Platform Standard Edition 5.0, API Specification,* Sun Microsystems, 2005.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.105-113
ISBN 83-922646-0-6


# Grouping and Joining Transformations in Data Extraction Process

Marcin Gorawski[1], Paweł Marks[1]

[1] Silesian University of Technology, Institute of Computer Science,
ul. Akademicka 16, 44-100 Gliwice, Poland
{Marcin.Gorawski, Pawel.Marks}@polsl.pl

**Abstract.** In this paper we present a method of describing ETL processes (Extraction, Transformation and Loading) using graphs. We focus on implementation aspects such as division of a whole process into threads, communication and data exchange between threads, deadlock prevention. Methods of processing of large data sets using insufficient memory resources are also presented upon examples of joining and grouping nodes. Our solution is compared to the efficiency of the OS-level virtual memory in a few tests. Their results are presented and discussed.

## 1 Introduction

Nowadays data warehouses gather tens of gigabytes of data. The data, before loading to the warehouse, is often read from many various sources. These sources can differ in terms of a data format, so there is necessity of applying proper data transformations making the data uniformly formatted. In consecutive steps the data set is filtered, grouped, joined, aggregated and finally loaded to a destination. The destination can be one or more warehouse tables. A whole process of reading, transforming and data loading is called data extraction process (ETL).

The transformations used in the ETL process can differ in terms of complexity. A few of them are simple (e. g. filtration, projection), whereas others are very long lasting and require a lot of operational memory (e. g. grouping, joining). However, the common feature of the transformations is that each one contains at least one input and an output. This allows to describe the extraction process using a graph, whose nodes correspond to objects performing some operations on tuples, and its edges define data flow paths.

Most of commercial tools like Oracle WB do not consider internal structure of transformations and graph architecture of ETL processes. Exceptions are researches [7, 8], where the authors describe ETL ARKTOS (ARKTOS II) tool. It can (graphically) model and execute practical ETL scenarios, providing us primitive expressions that brings under the control over the typical tasks using declarative language. Work [1] presents advanced research on the prototypes containing AJAX data cleaning tool.

To optimize the ETL process, there is often designed a dedicated extraction application, adjusted to requirements of a particular data warehouse system. Basing on authors experiences [4, 5], it was made a decision of building a developmental ETL environment using JavaBeans components. Similar approach was proposed in the meantime in work [2]. J2EE architecture with ETL and ETLLet container was presented there, providing efficient ways of execution, controlling and monitoring of ETL process tasks for continuous data propagation case.

Further speeding up of the ETL process forced us to give the JavaBeans platform up. An ETL-DR environment [3] is a successor to the ETL/JB and DR/JB [6]. It is a set of Java object classes, used by a designer to build extraction applications. These are analogous to JavaBeans components in DR/JB environment. However, object properties are saved in an external configuration file, which is read by an environment manager object. It frees us from recompilation of the application each time the extraction parameters change. In comparison to the ETL/JB and the DR/JB we significantly improved the processing efficiency and complexity of the most important transformations: grouping and joining. The possibility of storing data on a disk was added when the size of the data set requires much more memory than it is available.

In the following sections we present in details a method of describing ETL processes using graphs and we show how this description influences the implementation. The problems resulting from the graph usage are also discussed and methods of data processing using insufficient memory resources are presented.

## 2 Extraction Graph

Operations performed during the extraction process can be divided into three groups:

a) reading source data,
b) data transformations,
c) writing data to a destination.

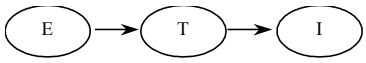



**Fig. 1.** One of the simplest extraction graphs. Node *E* is an extractor, node *T* is a transformation and node *I* is an inserter

Nodes belonging to the mentioned above operation groups are respectively: extractors (E), transformations (T) and inserters (I). From the graph's point of view extractors have only outputs, transformations have both inputs and outputs, whereas inserters contain inputs only. By connecting inputs to outputs we create a connection net that defines data flow paths (Fig. 1). The data flow inside the node is possible in one direction only: from the inputs to the outputs, in the opposite direction it is forbidden. It is also assumed that connection net does not contain closed loops, which means there is no possibility to enter the same graph node traversing along the selected path of the graph. Such a net of nodes and connections is called *directed acyclic graph* DAG.

## 3 ETL-DR Data Extraction Environment

The ETL-DR is our research environment designed in Java. It uses the extraction graph idea presented above to describe the extraction processes. During processing each graph node is associated with a thread, that is an instance of a transformation, or an extractor, or an inserter.

Available components are:

- extractors
  - FileExtractor (FE) – reads tuples from a source file,
  - DBExtractor (DE) – reads tuples from a database,
- transformations
  - AggregationTransformation (AgT) – aggregates a specified attribute,
  - FilterTransformation (FiT) – filters the stream of tuples,
  - FunctionTransformation (FuT) – user-definable tuple transformation,
  - GeneratorTransformation (GeT) – generates ID for each tuple,
  - GroupTransformation (GrT) – grouping,
  - JoinTransformation (JoT) – joining,
  - MergeTransformation (MeT) – merges two streams of tuples,
  - ProjectionTransformation (PrT) – projection,
  - UnionTransformation (UnT) – union,
- inserters
  - FileInserter (FI) – writes tuples to a destination file,
  - DBInserter (DI) – writes tuples to a database table via JDBC interface,
  - OracleDBInserter (ODI) – writes tuples to a database using Oracle specific SQL*Loader,
- specials
  - VMQueue (VMQ) – FIFO queue which stores data on a disk.

Most of the components process data on-the-fly, which means each tuple just received is transformed or analyzed independently and there is no need to gather a whole data set. The exceptions are: joining node JoT, grouping node GrT and VMQ queue.

### 3.1 Implementation of Graph Nodes Interconnections

In order to facilitate analysis of interconnections between the graph nodes we have to describe the structure of inputs and outputs of the ETL-DR extraction graph nodes. Each node has a unique ID. Each node input contains ID of a source node assigned by the graph designer, and an automatically assigned number of an output channel

of the source node. A node output is a multi-channel FIFO buffer with the number of channels equal to the number of inputs connected to the node (Fig. 2). When a node produces output tuples, it puts them into its output, where they are grouped into tuple packets. Upper limit of the packet size is defined by the designer. Packets are gathered in queues, separately for each output channel. The queue size is also limited to avoid unnecessary memory consumption.

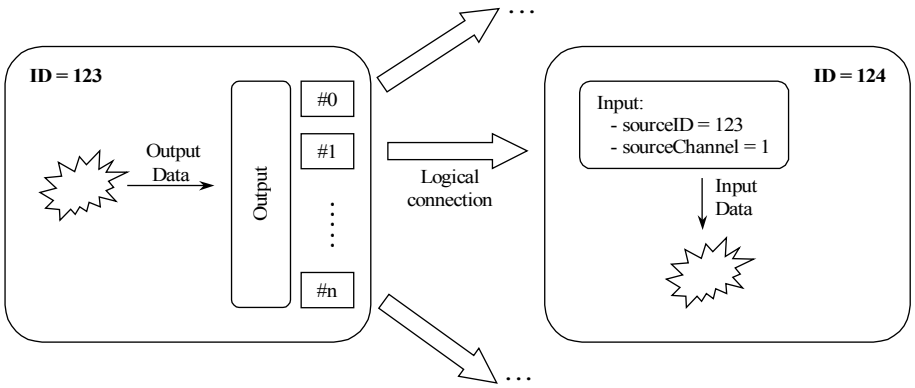



**Fig. 2.** Nodes interconnection on the implementation level. Data produced by the node 123 are stored in a multi-channel output buffer. Source of the node 124 is defined as a node with ID = 123 and logical channel number = 1

### 3.2 Data Exchange Between Nodes and a Risk of Deadlock

Let us analyze a case of processing performed by a part of the graph presented in Fig. 3a. The function node FuT(11) produces tuples with attributes *(eID, date, transactionsPerDay)*, and the grouping node GrT(12) computes average number of transactions for each employee. This is similar to the SQL query below:

```
SELECT eID, AVG(transactionsPerDay) AS avgTPD
FROM GrT_FuT

GROUP BY eID
```

The joining node JoT(13) performs an action defined by the following SQL query:

```
SELECT s1.eID, s1.date, s1.transactionsPerDay, s2.avgTPD
FROM JoT_FuT s1, JoT_GrT s2
WHERE s1.eID = s2.eID
```

Such simple operations like grouping and joining are dangerous because they can be a reason of deadlock. This is a result of the data transferring method between node threads which is used.

Joining node works as follows: it receives tuples from the slave input and puts them into a temporary buffer, next it receives tuples from the primary input. Each tuple from the primary input is checked if it can be joined with tuples in a temporary buffer according to the specified join condition. In the presented example, slave input is the one connected to the grouping node, because it is highly possible, that after grouping the size of the data set will decrease and less number of tuples will be kept in memory.

Tuples generated by the function node are simultaneously gathered in both output channels of the node for nodes JoT(13) and GrT(12). The grouping node aggregates data all the time, but the joining node waits for the grouped data first, and it does not read anything from the function node yet. After exceeding the limit of the output queue size, the function node is halted until the queue size decreases below the specified level. This way a deadlock occurs:

- the node FuT(11) waits until the node JoT(13) starts reading data from it,
- the node GrT(12) waits for the data from the node FuT(11),
- the node JoT(13) waits for the data from the node GrT(12).

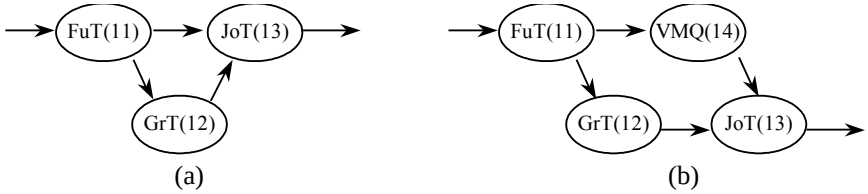



**Fig. 3.** Typical deadlock prone graph nodes connections (a) and a way of deadlock avoidance by use of VMQueue component (b)

To eliminate the reason of the deadlock we have to assure that the data from the function node FuT(11) are fetched continuously without exceeding the queue size limit. To do it we created a special VMQueue component. This is a FIFO queue with ability of storing data on a disk. It reads tuples from its input, no matter if they can be hand further or not. If tuples are fetched from VMQ node continuously it does nothing more but transfers data from the input to the output. In other case, it writes tuples to the disk in order to avoid overfilling of the output queue of its source node. Next, when VMQueue destination continues processing, the tuples are read from the disk and sent to the queue output. Inserting VMQueue node between FuT(11) and JoT(13) avoids the deadlock (Fig. 3b).

### 3.3 Formal Definition of the Deadlock Prone Graph Nodes Subset

A deadlock may occur if two or more data flow paths that split in one node of the graph, meet again in another node. In other words, a given node *X* is connected with any of its direct or indirect source nodes by two or more paths. This let us conclude that node *X* must have more than one input.

Let us represent a set of source nodes of the node *X* as *SourceNodes(X)*, and a set of source nodes of the *i*-th input of X as *InputSourceNodes(X,i)*. We can define:

- $InputSourceNodes(X,i)=SourceNodes(X.in[i].sourceID)\cup\{X.in[i].sourceID\}$
- $SourceNodes(X)=\varphi$ if *X* is an extractor,
  $InputSourceNodes(X,i)$¿ if *X* is a transformation
  or an inserter,
- $CommonNodes(X,i,j)=InputSourceNodes(X,i)\cap InputSourceNodes(X,j)$ ,
- $LastNode(N)=\{X\in N:SourceNodes(X)\}=N\{X\}$ .

If for each node *X* of the extraction graph, which is not an extractor, the following condition is satisfied:

$$i\neq j\Rightarrow CommonNodes(X,i,j)=\varphi$$¿

then deadlock cannot occur. Otherwise deadlock is possible and we should use VMQueue component and insert it into the graph, to avoid the application hang. Insertion of VMQueue node makes sense only behind the nodes from a $LastNode(CommonNodes(X,i,j))$ set, that is a set of the last nodes from the set of common parts of the two data flow paths. In the example presented in the previous section it was the FuT(11) node (Fig. 3b).

### 3.4 Temporary Data Buffering on Disk

During an extraction process a large number of tuples is processed. When they need to be buffered, there is a problem of selection of the right place for the buffer. Keeping them in memory is impossible because the size of the data set is usually much bigger than the size of the available RAM. The only solution is storing the data on a disk. Two approaches are possible: virtual memory supported by the operating system or storing implemented on the application level in algorithms used in transformation nodes. In our ETL-DR environment the nodes using application-level virtual memory are: VMQueue, GroupTransformation and JoinTransformation.

#### 3.4.1 VMQueue Component

As it was presented in section 3.2 VMQueue component is a FIFO queue able to store the buffered data on a disk. Its task is to ensure the data is read from its source as it comes, even if the node receiving data from

VMQueue does not work. In such a case tuples are stored in a disk file rather than put into the output buffer. Next when possible, tuples are read from the file and hand further. Because of a sequential access to the disk file, this solution is more efficient than the OS-level virtual memory.

### 3.4.2 GroupTransformation Component

A grouping component can work in one of three modes:

a) input tuples are sorted according to the grouping attribute values,
b) tuples are not sorted, grouping in memory,
c) tuples are not sorted, external grouping.

In case a) aggregates are computed as they come, and memory usage level is very low. In case b) each new combination of the grouping attributes is saved in a hash table with associated aggregates. If such a combination appears again during processing, it is located and the aggregates are updated. The number of entries in the hash table at the end of the processing equals to the number of tuples produced. Both cases a) and b) use only RAM.

```
procedure Group()
Begin
  List fileList;
  While Input.hasTuples() do
    Tuple T = Input.getTuple();
    If not HM.contains(Attributes(T)) then
      HM.put(Attributes(T), Aggregates(T));
    End if
    Aggregates AG = HM.get(Attributes(T));
    AG.doAggregate(T);
    If HM.size() > SIZELIMIT then
      fileList.add(WriteToFile(HM));
      HM.clear();
    End if
  End while
  AggrSource as = getSource(fileList, HM);
  Aggregates AG = null;
  While as.hasNext() do
    If AG == null then
      AG = a.next();
    Else
      Aggregates newAG = as.next();
      If (newAG.attr == AG.attr) then
        AG.aggregate(newAG);
      Else
        ProduceOutputTuple(AG);
        AG = newAG;
      End if
    End if
  End while
  ProduceOutputTuple(AG);
End
```

**Fig. 4.** External grouping algorithm

Case c) has features of the processing used in cases a) and b). First, data set is gathered in the hash table and aggregates are computed. When the number of entries in the table exceeds the specified limit, the content of the table is written to the external file in sorted order according to the grouping attribute values. Next, the hash table is cleared and the processing is continued. Such a cycle repeats until the input tuple stream ends. Then the data integration process is run. Tuples are read from the previously created files and final aggregates values are computed. This is very similar to case a) processing with the exception of getting data from external files instead of the node input.

### 3.4.3 JoinTransformation Component

A joining node works basing on the algorithm presented in Fig. 5. The first step is collecting tuples from the slave input. They can be loaded to a temporary associating array or written to a temporary disk file. Before writing to the file, tuples are sorted according to the joining attributes using external version of the standard Merge-Sort algorithm: tuples are gathered in memory, if the limit of tuples in memory is exceeded they are sorted and written to a file. Next portions of the data set are treated in the same way. Finally tuples from all the generated sorted files are integrated into one big sorted file. Sorting let us to locate any tuple in the external file

in *log(n)* time using the binary search algorithm. Additional indexing structure located in memory decreases searching time also, by reducing the number of accesses to the file. The index holds locations of the accessed tuples, which enables narrowing down the searching range when accessing consecutive tuples.

```
procedure Join()
Begin
  While Input(2).hasTuples() do
    Tuple T = Input(2).getTuple();
    HM.put(Attributes(T), T);
  End while
  While Input(1).hasTuples() do
    Tuple T = Input(1).getTuple();
    Tuple[] TT = HM.get(Attributes(T));
    For each JT in TT do
      Tuple O = Join(T, JT);
      ProduceOutputTuple(O);
    End for
  End while
End
```

**Fig. 5.** General join algorithm

The second phase is the same, no matter if the temporary buffer is located in memory or on a disk. Only the implementation of the HM (HashMap) object changes in the algorithm presented in Fig. 5. Each tuple from the primary input is checked if it can be joined with tuples in the temporary buffer according to the specified join condition.

## 4 External Processing Tests

For tests we used data files that forced Java Virtual Machine to use much more memory than it was physically available. Tests were performed on the computer with AMD Athlon 2000 processor working under WindowsXP Professional. During tests we were changing the size of the available RAM.

### 4.1 Grouping Test

Grouping was tested basing on the extraction graph containing an extractor *FE*, a grouping node *GrT* and an inserter *I* (Fig. 6). The extractor reads the tuple stream with attributes *(eID, date, value)*, in which for each employee *eID* and for each day of his work, his transaction values were saved. The number of employee transactions per day varied from 1 to 20. The processing can be described by SQL query as:

```
SELECT eID, date, sum(value) as sumVal,
       count(*) as trCount
FROM GrT_FE
GROUP BY eID, date
```

The processing time was measured depending on the number of input tuples (10, 15, 20 and 25 millions) and the type of processing. The result chart contains total processing time (TT) and the moment of loading the first tuple to a destination, so called *Critical Time* (CT). During all the tests using external grouping (Ext) JVM was assigned only 100MB of RAM. During grouping in memory, we examined two cases: JVM memory was set with some margin (Normal) and with minimal possible amount of RAM (Hard) that guaranteed successful completing of the task. The results we obtained are shown in Fig. 7.

The test computer contained 384MB physical RAM, and for JVM using virtual memory and for 10m and 15m tuples it was assigned respectively 450MB and 550MB during Normal test, then 300MB and 425MB during Hard test.


**Fig. 6.** Grouping test extraction graph

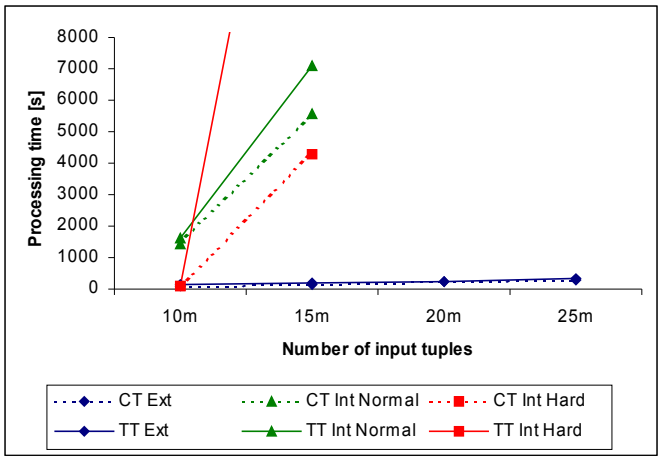



**Fig. 7.** Processing times measured during grouping test. TT is a total processing time, whereas CT denotes a moment when the first output tuple is produced (*Critical Time*)

As it can be seen, the most efficient processing method is definitely the one using application-level data storing. Its processing time changes from 129 sec. to 322 sec. depending on the number of input tuples. The use of OS-level virtual memory causes that the whole process takes much more time. Only for 10 millions of input tuples and strongly limited JVM memory, what resulted in a very low usage of the virtual memory, we obtained results minimally better than for built-in data storing. However, for 15 million tuples the processing takes an extremely long time (the line going rapidly outside the chart). The main reason of so low efficiency of a virtual memory are random accesses to the memory caused by updating aggregates in temporary buffers and Java garbage collector. The application-level storing accesses data files sequentially, and it causes this method is much more efficient.

We did not finished OS-level virtual memory tests for 20m and 25m tuples because it was taking extremely long time (several hours). Our goal was only to show that the application-level buffering can be much better than OS-level buffering.

### 4.2 Joining Test

Joining test is based on the extraction graph shown in Fig. 8. The extractors read the same number of tuples: *FE1* reads tuples with attributes *(eID, date, depID)*, describing where each employee was working each day, whereas *FE2* reads a set of tuples produced in the previous test *(eID, date, sumVal, trCount)*. Joining attributes are *(eID, date)* and processing times were measured for the following number of input tuples from each extractor: 10, 15 and 20 millions.

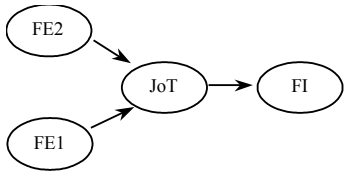



**Fig. 8.** Joining test extraction graph

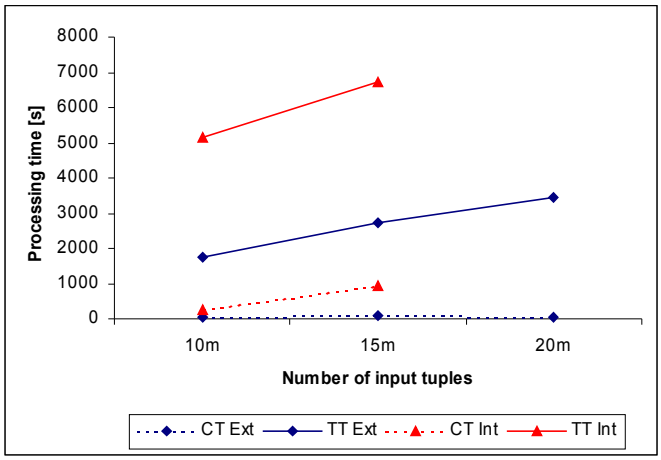



**Fig. 9.** Processing times measured during joining test. TT is a total processing time, whereas CT denotes a moment when the first output tuple is produced (*Critical Time*)

During the test the computer was equipped with 256MB RAM, JVM was assigned 100MB RAM when joining with data storing on disk was used, and respectively 400MB and 600MB for 10 and 15 million of tuples when using virtual memory. In this test we can still observe benefits of using application-level data storing, but the difference in comparison to OS virtual memory is not so big as in the grouping test because this time the external file is accessed randomly, not sequentially. The results we obtained are presented in Fig. 9.

### 4.3 Real Extraction Test

We also performed a real extraction test. The ETL process generates a star schema data warehouse containing a fact table and two dimensions. In this test, both grouping and joining nodes appear in the extraction graph and they run concurrently: when the grouping node GrT(2) produces output tuples, the joining node JoT(30) puts them into its internal buffer (memory or a disk file). This test let us examine the behavior of the buffering techniques when more than one node require a lot of memory resources.

Size of the input data set was 300MB. JVM required 475MB RAM to complete the task using virtual memory, and only 100MB when using application-level data storing. The computer had 256MB RAM. The ETL process using data storing took only 26 minutes, whereas when using the virtual memory, it needed 3 hours to complete only 10% of the whole task (the whole processing could take even 30 hours). Continuation of the test did not make sense, because we could already conclude that in this case the efficiency of the virtual memory was extremely low.

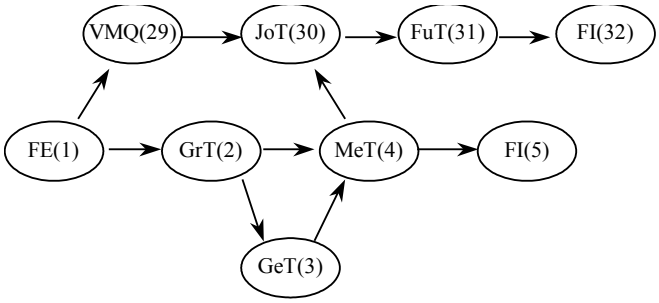



**Fig. 10.** The main part of the extraction graph generating star schema data warehouse. Path FE(1)-FI(32) generates fact table, whereas path FE(1)-FI(5) is responsible for producing one of the dimension tables. Extractor FE(1) reads 300MB data file

We think that the obtained results are the result of the random accesses to the VM swap file. When many nodes keep a lot of data in a virtual memory and access it randomly (because each node run as an independent

thread) the swap file has to be read and written very often from various locations. This does not take place during application-level buffering, the external files are accessed sequentially if only it is possible (depending on the algorithm that is used).

## 5 Conclusions

In this paper we presented a concept of describing extraction processes using graphs, the meaning of graph nodes and the graph edges in the extraction process. We focused on a few implementation aspects like interconnections between nodes and the possibility of deadlock occurrence when particular graph structures are used. A method of avoiding deadlocks was also presented and it was described by mathematic formulas. Next we introduced algorithms for external data queuing, groping and joining.

Although not tested in this paper, the presented data queuing is the efficient method of avoiding deadlocks that may occur in our ETL-DR extraction environment due to the data transferring method we used. The grouping transformation can process data sets of any size, the only limitation is the available temporary disk space. It makes use of the additional tuple stream properties, such as sorted order according to the values of the grouping attributes. The joining transformation also can process unlimited number of tuples. It can store its slave-input tuples to disk files in a sorted order and then access any tuple in the file in $\log(n)$ time.

Our research proves that a virtual memory offered by operating systems is not always the efficient solution. Dedicated algorithms of storing data in external files working on the application level are more efficient due to elimination of random accesses to a disk, which is the weakest side of the OS virtual memory. This weakness is especially emphasized in Java applications. A typical JVM prefers allocating new memory blocks to freeing unnecessary ones as soon as possible. This may be very efficient when only physical RAM is in use, but when JVM enters virtual memory area and a garbage collector tries to recover unused memory blocks from it, the efficiency of a whole application dramatically drops.

## References

1. Galhardas H., Florescu D., Shasha D., Simon E.: Ajax: *An Extensible Data CleaningTool*, In Proc. ACM SIGMOD Intl. Conf. On the Management of Data, Teksas (2000).
2. Bruckner R., List B., Schiefer J.: *Striving Towards Near Real-Time Data Integration for Data Warehouses*, DaWaK 2002
3. Gorawski M., Marks P.: *High Efficiency of Hybrid Resumption in Distributed Data Warehouses*, HADIS 2005.
4. Gorawski M., Piekarek M.: *Development Environment ETL/JavaBeans*, Studia Informatica, vol. 24, **4** (56), 2003.
5. Gorawski M., Siódemak P.: *Graphic Design of ETL Applications. Studia Informatica*, vol. 24, **4** (56), 2003.
6. Gorawski M., Wocław A., *Evaluation of the Efficiency of Design-Resume/JavaBeans Recovery Algorithm.* Archives of Theoretical and Applied Informatics, **15** (1), 2003.
7. Vassiliadis, P. A. Simitsis, S. Skiadopoulos. *Modeling ETL Activities asGraphs*, In: Proc. 4th Intl. Workshop on Design and Management of Data Warehouses, Canada, (2002).
8. Vassiliadis P., Simitsis A., Georgantas P., Terrovitis M.: *A Framework for the Design of ETL Scenarios*, CAiSE 2003.

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 115–120 

# The Use of Self-Organising Maps to Investigate Heat Demand Profiles

Maciej Grzenda

Warsaw University of Technology, Faculty of Mathematics and Information Science,
Pl. Politechniki 1, 00-661 Warszawa
grzendam@mini.pw.edu.pl,
WWW home page: http://www.mini.pw.edu.pl/grzendam

**Abstract.** District heating companies are responsible for delivering the heat produced in central heat plants to the consumers through a pipeline system. At the same time they are expected to keep the total heat production cost as low as possible. Therefore, there is a growing need to optimise heat production through better prediction of customers needs. The paper illustrates the way neural networks, namely self-organised maps can be used to investigate long-term demand profiles of consumers. Real-life historical sales data is used to establish a number of typical demand profiles.

## 1 Introduction

Modern utility companies transmitting heat to the consumers face new challenges. The development towards deregulated and more competitive energy markets means that many consumers may decide to resign from district heating services based on central heat sources and replace them with decentralised or individual boiler systems. In order to tackle this issue, utility companies have to improve service quality and minimise the price at the same time. For a broad overview of district heating see [8].

It is worth emphasising that the most significant part of district heating cost is usually the cost of producing the heat. Thus, by optimising heat supply significant cost reduction can be obtained. However, the latter objective can not be fulfilled without detailed analysis of consumers' demand profiles.

The work investigates yearly heat demand profiles and continues previous research on power demand prediction [2, 3]. The objective is to devise a set of typical heat demand profiles that would correspond to typical groups of consumers. By obtaining such predicted yearly demand profiles, long-term optimisation of heat supply can be attempted. Still, the investigation of daily load profiles is also required to answer the needs of short-term optimisation.

The paper presents the way self-organising maps [4, 5] can provide valuable insight into typical heat demand profiles. The information available in sales database system has been retrieved and transformed into historical data. The time series [1] of sales volume for each consumer has been obtained. As the heat sales strongly depends on space heating demands it varies in accordance with time of the year and average temperatures.

Finally, self-organised maps have been used to build a network of neurons representing typical sales profiles. The work concentrates on the application of neural networks to investigate these typical heat sale patterns.

The remainder of this paper is organised as follows:

- section 2 describes in details the data set of the problem,
- neural networks-based approach is outlined in section 3,
- the results of the computations are discussed in section 4,
- section 5 summarises the whole work.

## 2 Data set

The problem analysis is based on the data set from one of the Polish district heating utilities. The sales information from the billing system for the period of last four years has been used. For each consumer

average monthly heat sale has been calculated. The heat sales information is the total heat consumption measured for each consumer separately. It is worth noting here that sales information is available for every single consumer. At the same time information regarding the type of a building, its technical attributes, number of rooms incl. rooms for rent and its thermal properties are available to different extent and for some consumers only. Therefore, to build up preliminary prediction models sales data should be used first and then for selected consumers with the same overall demand profile, detailed prediction models can be devised and tested.

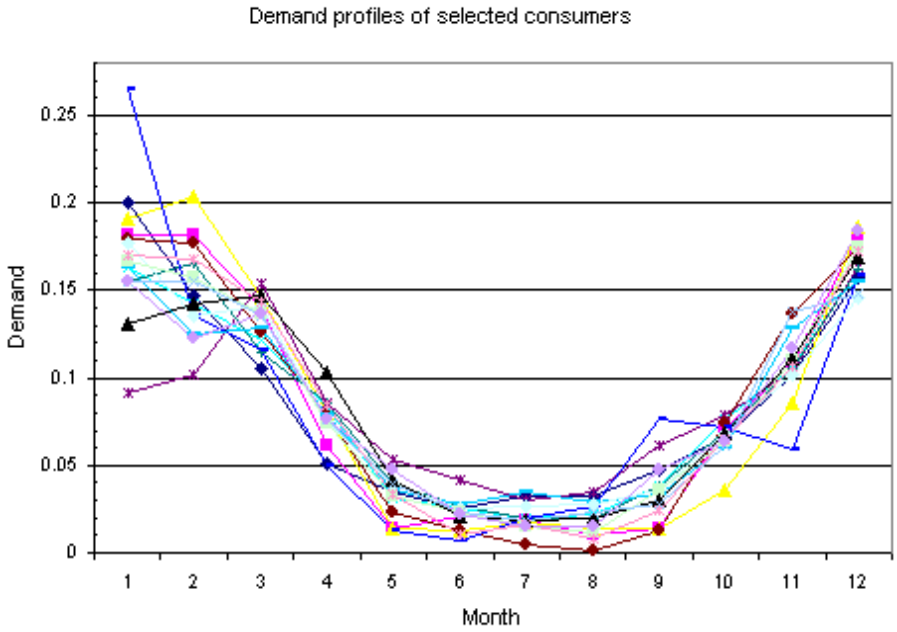



**Fig. 1.** Sample demand profiles $D_i$ of different consumers

It is worth emphasising that diverse thermal energy needs of consumers are answered by district heating system. Not only do they include space heating necessary to ensure human comfort, but also domestic hot water and industrial needs. Thus, while prevailing majority of demand is due to space heating requirements, other factors may strongly affect demand profiles and result in different demand profiles for different categories of consumers. Still, if the weather conditions were the only factors to influence heat demand, normalised profiles of different consumers would be the same.

Proper identification of consumers' needs in long term is necessary for overall heat demand prediction. Without such prediction, neither volume of sales nor volume of heat supply can be estimated. However, a district heating utility must assess both of them to evaluate the share of heat sources with different costs in the overall sales of heat. Moreover, in order to predict the impact of connecting new consumer to the system, its demand profile must be taken into account. The latter profile is supposed to answer the question whether the impact of connecting for instance a new hospital to the system is different in terms of normalised profile than the impact of connecting a new hotel.

In addition, the construction of prediction models can be treated as a task of creating a number of distinctive models for different consumers groups not to try to capture the behaviour of a school, a hotel, office, family house and a restaurant in the same model. All the same, the latter objective can not be fulfilled without understanding of typical demand profiles and their relation to consumer categories.

After detailed analysis $N = 616$ sales profiles have been obtained. Every sales profile of consumer $i, i = 1, \ldots, 616$ is represented by a vector $S_i = (s_{i,1}, \ldots, s_{i,12})$, where $s_{i,m}$ denotes the average heat sale to consumer $i$ during month $m$. Months are indexed in a traditional way i.e. 1 stands for January.

Furthermore, each sales profile has been normalised so as to obtain average demand profile $D_i = (d_{i,1}, d_{i,2}, \ldots, d_{i,12})$ out of original consumer profile $S_i$ as follows:

$$d_{i,k} = \frac{s_{i,k}}{\sum_{m=1}^{m=12} s_{i,m}}, k = 1, \ldots, 12 \quad (1)$$

Thus, $N = 616$ normalised demand profiles have been obtained. These correspond to the average yearly consumption of heat at each consumer. Some of demand profiles are depicted on fig. 1.

While, in general, demand profiles reflect space heating needs resulting from average air temperatures, one can observe that minimal demand is reached by the consumers in the period between May and August. Similarly, peak heat consumption is attained between December and March. In other words, the exact time location of minimal and maximal heat consumption significantly varies among consumers. What is of outstanding importance even normalised monthly heat sales can differ as much as 300% among sample consumers randomly selected from the whole data set. These statements are further supported by statistic data analysis.

Fig. 2 depicts standard deviation of monthly demand $D_{i,m}, i = 1, \ldots, N$. It can be easily seen that the diversity among consumers in terms of heat demand reaches its peak in August, December, January and February and remains relatively low during remaining part of the year. This diversity may be due to the fact, there are many hotels and guest houses with rooms for rent among heat consumers. As a consequence, demand for heat is increased during high season i.e. in December, January, February, July and August. Therefore, whether one average demand profile could be used to approximate demands of all the consumers remains an open issue. However, if a number of different average load profiles could be devised, then they should differ most during the months listed above i.e. during the months with the largest standard deviation of $D_{i,m}, i = 1, .., N$.

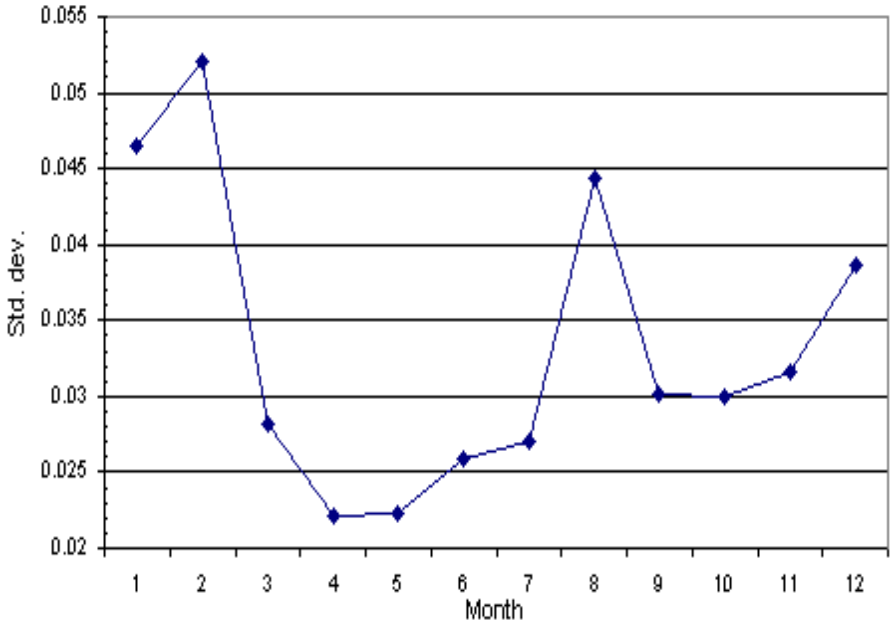



**Fig. 2.** Standard deviation of consumer's average demand during different months of the year.

## 3 Neural networks and demand profile identification

While a separate demand profile $D_i$ for each consumer could be devised, the problem remains to identify common profile groupings. The latter groupings are necessary for further market analysis, for instance to

evaluate the impact of new consumers on the district heating system in view of prospective heat demand. The question is whether observed variety of demand profiles can be explained by the existence of a number of diverse demand profiles among consumers.

In order to address this issue, self-organising neural networks have been applied. The purpose of using self-organised neural networks, namely self-organised maps (SOMs) [5], is to provide measure of locating demand pattern groupings i.e. typical profiles $\hat{D}_k, k << N$ that would explain diversity in the population of $D_i, i = 1, .., N$ demand profiles. The following solutions have been applied:

- two-dimensional square lattice of $J$ neurons has been applied,
- demand profiles $D_i, i = 1, .., N$ have been applied as input patterns.

Therefore, each neuron $j$ located in the lattice is represented by a weight vector

$$w_j = [w_{j,1}, w_{j,2}, .., w_{j,12}]^T, j = 1, 2, \ldots, J. \tag{2}$$

During the learning process neurons are tuned to the input patterns. Moreover, the weight update algorithm tunes the weights of the winning neurons and some of its spatially close neighbour neurons. Thus, topographical map of input space is being created [4]. In other words, as a result of this cooperative process, both input pattern groupings can be identified and their relation. Therefore, SOM-based analysis is sometimes treated as a generalised form of standard Principal Component Analysis (PCA) [6].

## 4 Results

A number of computation series have been performed, using different weight update algorithms. Not only standard Winner-Takes-All (WTA) [4, 5], but also Winner Takes All with Conscience (CWTA) [4] and Neural Gas (NGAS) [7] algorithms have been used to tune neuron weights. The latter algorithm aims to overcome deficiencies of standard WTA algorithm and has been used to construct self-organised maps described below. Among these deficiencies of standard WTA weight update algorithm, overrepresentation of regions with low input density is not of least importance [4].

Distinctive features of NGAS algorithm are:

- the neighbourhood function defined as follows:

$$G(j, D_i) = \exp(-\frac{m(j)}{\lambda}), j = 1, \ldots, J \tag{3}$$

  where $m(j)$ defines the index of neuron $j$ in the sequence of neurons ordered in accordance with growing distance of neuron $j$ from pattern $D_i$ i.e. $\parallel w_j - D_i \parallel$.
- a number of neurones are updated in each step, however due to the $G(j, x)$ definition it is enough to update only first $K$ neurons in the sequence. The latter simplification results in substantial reduction of necessary computations comparing to standard algorithm.

In all cases, for each neuron $j, j = 1, \ldots, J$ in the lattice, at the end of tuning phase, the winning count $Y_j$ has been calculated in the following manner:

$$Y_j = \sum_{i=1}^{N} \mathrm{eval}\,(i, j) \tag{4}$$

while

$$\mathrm{eval}\,(i, j) = \begin{cases} 1 & \text{if } \parallel D_i - w_j \parallel = \min_{k=1,\ldots,J} \parallel D_i - w_k \parallel \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Thus $Y_j : j = 1, \ldots, J$ denotes the number of input patterns that are the most similar to the weight vector $w_j$ in terms of Euclidean distance.

For 49 out of $J = 100$ neurons, $Y_j = 0$. On the other hand, a number of neurons wins frequently. Fig. 3 depicts weight vectors $w_j : Y_j > 20$. Each profile corresponds to a single neuron. Its $(x, y)$ location on the neuron lattice is provided in the figure legend.

Obtained results show that a number of distinctive demand profiles providing centroids of input consumer's profiles have been identified. This suggests that observed diversity in the input patterns can be explained by underlying typical demand profiles. In addition, the fact that some 50% of neurones remain inactive, in spite of using NGAS algorithm that promotes diversity among neurones, points that significant simplifications can be made. In other words, when dealing with heat demand prediction, it is enough to concentrate on a limited number of consumers demand patterns.

The profile groupings identified in the form of weight vectors reflect the diversity of sample demand profiles illustrated on fig. 1. In particular, one may notice that peak consumption is reached in January, February or December, depending on resulting $\hat{D}_k$ profile considered. Thus, typical demand profiles received from self-organising map reflect diversity of original demand profiles.

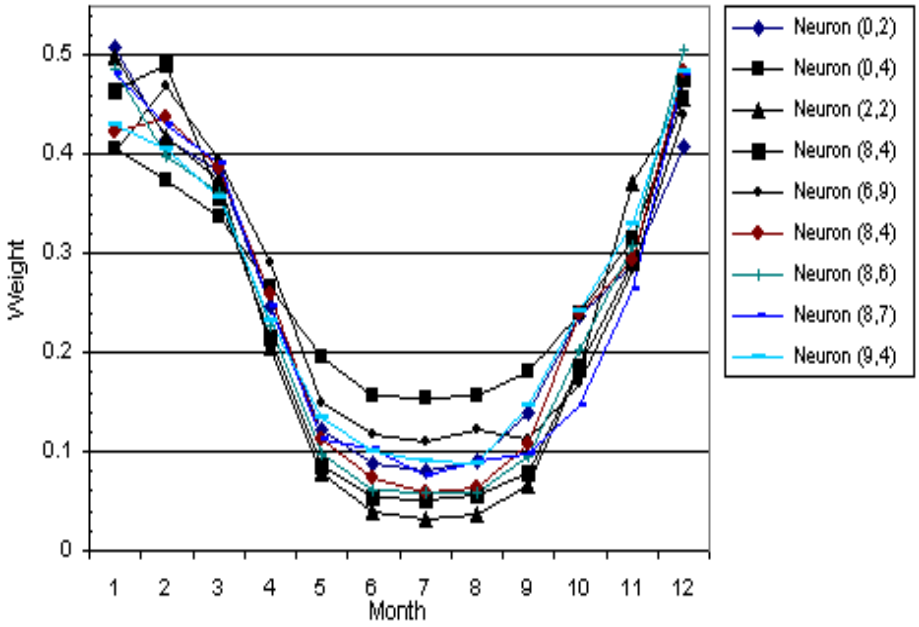



**Fig. 3.** Demand profiles represented by weight vectors of the winning neurons.

In order to investigate the relation between consumer category and its typical demand profile, additional computation series have been performed. For each neuron $j$ the total number of closest consumers' profiles $Y_j = \sum_{c=1}^{C} Y_j(c)$ has been calculated. $c = 1, \ldots, C$ denotes the index of a consumer category and $C$ denotes the total number of categories used. Sample categories are family houses (FH), commercial buildings (C), hotels (H), hospitals (HS). The following conclusions have been made when analysing the number $Y_j(c)$ for the neurons of the lattice:

- hospital demand patterns are closest to the two neighbouring neurons on the lattice,
- more than 80% of hotel consumers can be related to just 4 neurons grouped in a distinctive spatial cluster on the lattice,
- also commercial buildings form a spatial cluster of 6 neurons on the lattice, however some 35% of these consumers can not be related to this cluster,
- demands of family houses are very different, the neuron with highest win rate for FH consumers represents only 8.9% of all FH consumers.

As a consequence, the SOM-based demand analysis suggests that some groups of consumers, like hospitals, hotels and to some extent commercial buildings share similar demand profiles. On the other hand, some new subcategories for FH consumers would need to be established. Still, one may investigate different subgroups of consumers associated with the neurons of the lattice and take into account their spatial relation to other

consumers. For instance, the fact some FH resemble hotels may be due to the fact that some of these consumers offer many rooms for rent, thus increase their demand for heat significantly during high season. Consequently, the context map created by assigning $Y_j(c)$ values to the neurons helps understand the relation between different consumer classes. In addition, it can be used by the marketing department to analyse and select the best matching demand pattern for a new consumer basing on its category and other attributes. As a consequence, the impact of a new consumer on the district heating system in terms of averaged demand profile can be evaluated and used for overall demand and supply analysis.

## 5 Conclusions

The paper shows the way SOM neural networks can provide for time-series data analysis. Results of the computation suggest that diversity of heat demand profiles can be explained by a number of different base demand profiles. A number of such base profiles has been obtained in the form of weight vectors of the neurons with the highest winning rate. The results strongly suggest that a limited number of typical demand profiles can be used to provide valid representation of all consumers. Moreover, the method helps validate existing consumers classes in view of consumers demand. It has been shown that for some consumer categories typical demand profile can be associated, while remaining categories require more detailed classification to ensure appropriate demand assignment.

Future works will include the identification process of consumer's profile through supervised learning based on descriptive attributes, like ordered heat volume, consumer's category (domestic, public use, company office, school etc.). In addition similar profiles will be created for short time consumption to answer the requirements of short-time prediction. Thus, hourly demand will be analysed.

## References

1. Brandt S., *Data Analysis. Statistical and Computational Methods for Scientists and Engineers,* Springer Verlag, (1999).
2. Grzenda, M., Macukow, B. *Evolutionary Model for Short Term Load Forecasting,* Proc. of MENDEL 2001 conference. Brno, pp. 119-124, (2001).
3. Grzenda,M. , Macukow, B. *Evolutionary Neural Network-Based Optimisation for Short-Term Load Forecasting,* Control and Cybernetics, **vol. 31**, 2/2002, pp. 371–382, (2002).
4. Haykin, S. *Neural Networks: a Comprehensive Foundation,* Prentice-Hall Inc., (1999).
5. Jang J.-S. R., Sun C.-T., Mizutani E., *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence,* Prentice Hall, (1997).
6. Lattin, J., Carroll, J., Green, P.: *Analyzing Multivariate Data,* Thomson Learning, (2003).
7. Martinetz M., Berkovich S., Schulten K., *"Neural-gas" Network for Vector Quantization and Its Application to Time Series Prediction,* Trans. Neural Networks, **vol.4**, pp. 558–569, (1993).
8. Pierce M. A., *Environmental Benefits from District Heating in the Nordic Capitals,* 16th Congress of the World Energy Council, Tokyo, (1995).

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.121-129
ISBN 83-922646-0-6


# Analiza możliwości identyfikacji ruchu postaci w oparciu o technikę Motion Capture

Bartosz Jabłoński[1], Ryszard Klempous[1], Damian Majchrzak[2]

[1] Politechnika Wrocławska, Instytut Informatyki, Automatyki i Robotyki,
ul. Janiszewskiego 11/17, 50-370 Wrocław, Polska
{Bartosz.Jablonski, Ryszard.Klempous}@pwr.wroc.pl
[2] Politechnika Wrocławska, KNS TRAF przy I-6
Damian.Majchrzak@gmail.com

**Streszczenie.** Ruch człowieka jest złożonym sygnałem, którego różne cechy charakterystyczne mogą posłużyć do analizy biometrycznej. Praca opisuje możliwości zastosowania analizy widma amplitudowego oraz metody *DTW* (*Dynamic Time Warping*) dla sygnałów rotacji dolnych kończyn w celu identyfikacji osoby. W wielu pracach zostało pokazane, że informacje zawarte w ruchu dolnych kończyn oraz bioder mogą być nieocenionym źródłem wiedzy na temat ruchu specyficznego dla każdej osoby. Za ich pomocą można rozróżnić oraz zidentyfikować zarejestrowany ruch człowieka. Dzięki rozwojowi przedstawionej analizy, możliwe będzie w przyszłości sprawne identyfikowanie ludzi na odległość. W pracy przedstawione zostały możliwości wynikające z zastosowania różnych metod analizy ruchów oraz zaproponowano kierunki rozwoju identyfikacji biometrycznej[1].

## 1. Wstęp

Biometria jest dziedziną przeżywającą gwałtowny rozwój. Szczególny nacisk kładziony jest na jej aspekty związane z poprawą bezpieczeństwa publicznego, a także na zastosowaniach medycznych. Biometria rozwija się w różnych kierunkach. Oprócz doskonalenia znanych metod do identyfikacji, takich jak analiza odcisków palców, otoskopia, porównywanie siatkówki i tęczówki oka oraz analiza głosu, rozwijane są nowe koncepcje. Powstają systemy wykorzystujące biometrię behawioralną bazujące na sposobie pisania, sposobie naciskania klawiatury komputera, a także na analizie ruchu różnych części ciała.

### 1.1 Rejestracja ruchu - Motion Capture

Systemy rejestracji ruchu są intensywnie rozwijane na potrzeby przemysłu rozrywkowego oraz medycyny. Konieczność ograniczania kosztów związanych z produkcją filmów zmusza do szukania nowych, lepszych sposobów tworzenia animacji komputerowych. Wielkim udogodnieniem w procesie animowania postaci jest zastosowanie materiału zgromadzonego przez systemy *Motion Capture* [16]. Systemy te opierają się na przechwytywaniu ruchu aktora, czyli jego rejestrowaniu i przetwarzaniu do postaci trójwymiarowych danych cyfrowych. Główną zaletą systemów przechwytywania ruchu jest ich dokładność. Pozwala ona na badanie wszelkich niuansów ruchu poprzez precyzyjne określenie pozycji poszczególnych segmentów ciała ludzkiego. Dokładność ta pozwala na próbę medycznego zastosowania tej techniki rejestracji ruchu [4]. Dane z systemów *MC* mogą dostarczyć cennych medycznych informacji np. na temat zaburzeń stanu równowagi, które mogą być wynikiem wczesnego stadium choroby Parkinsona lub stwardnienia rozsianego. Medycyna sportowa jest również obszarem, w których omawiana technika mogłaby przynieść obiecujące rezultaty. We wszystkich technicznych dyscyplinach sportu, gdzie ważna jest ergonomia oraz precyzja ruchu, technika *MC* byłoby idealnym narzędziem wspomagającym proces treningu sportowca. Jedyną barierą szerszego zastosowania tej metody rejestracji ruchu jest jej wysoki koszt i konieczność używania skomplikowanej rozbudowanej aparatury. Technika ta stanowi jednak źródło nieocenionych informacji na temat istoty ruchu. Informacje te mogą potem

1 Projekt częściowo finansowany ze środków *Fundacji na Rzecz Nauki Polskiej*.

z dobrym rezultatem być użyte w konwencjonalnych systemach bazujących na technice wideo i rozpoznawaniu ruchu na odległość.

### 1.2 Rejestracja wideo

Najpopularniejszym sposobem rejestracji ruchu jest nagrywanie za pomocą kamery wideo. Metoda ta jest najprostsza i najbardziej rozpowszechniona. Jednak jej zastosowanie do analizy ruchu przysparza najwięcej problemów. W zagadnieniach związanych z biometrią niezbędne jest posiadanie jak najdokładniejszych danych. Przekształcenie prostego obrazu wideo (nawet z wielu źródeł) w trójwymiarowy obarczone jest znacznym błędem. Ponadto technika ta jest szczególnie czuła na zakłócenia atmosferyczne. Niemniej jednak prowadzone są intensywne badania nad wykorzystaniem sekwencji wideo do rejestracji i analizy ruchu. Niskie koszty urządzeń do rejestracji oraz dalszy rozwój algorytmów przetwarzania obrazów mogą spowodować, że będzie ona wkrótce powszechnie stosowana. Należy zauważyć, że niektóre algorytmy opisywane w pracy mogą być, po odpowiedniej adaptacji, zastosowane do analizy danych z obrazów dwuwymiarowych.

### 1.3 Ruch segmentów ciała jako materiał analizy biometrycznej

W przypadku charakteryzacji tak założonego procesu, jakim jest ruch człowieka potrzebne jest wyselekcjonowanie jego najważniejszych cech mających zastosowanie biometryczne. Dobór tych cech jest sprawą kluczową, ponieważ na nich opierać się będzie cały proces identyfikacji. Jednym z wielu podejść do ruchu człowieka oraz jego analizy jest podejście traktujące ruch jako sygnał, a raczej jako zbiór sygnałów zmiennych w czasie. Takimi sygnałami mogą być rotacje poszczególnych kończyn bądź stawów względem stawów odniesienia (lub względem głównego stawu odniesienia tzw. *root*). Mogą nimi być także przyspieszenia, prędkości kończyn oraz pozycje w przestrzeni dwuwymiarowej bądź trójwymiarowej wybranych punktów ciała ludzkiego. Wiele własności ruchu oraz parametrów go opisujących jest podstawą różnorodnych metod biometrycznej analizy ruchu człowieka. Na podstawie istniejącej wiedzy, na potrzeby pracy, przeprowadzono szereg eksperymentów badających możliwość zastosowania różnych podejść do zagadnienia identyfikacji człowieka na podstawie jego ruchu.

Murray [10] postawił tezę, że ruch bioder oraz lędźwi różni się znacznie u poszczególnych badanych osób. W pracy [11] poddawane są analizie rotacje podudzia, drgania bioder oraz wychylenie kości guzicznej względem szyi. Podejście to obarczone jest takimi samymi problemami jak poprzednie. Biodra i szyja często skrywana jest pod ubraniem, przez co bardzo trudne jest odseparowanie tych punktów w przypadku rejestracji ruchu metodami innymi niż *MC*. Głównym segmentem ciała ludzkiego, którego ruch poddawany jest analizie są kończyny dolne. Jest to podejście zaproponowane w pracach [2, 3, 9, 14, 15]. Zaletą jego jest łatwość odseparowania ruchu uda i podudzia z obrazu dostarczanego przez kamery. W odróżnieniu od ruchu rąk, kończyny dolne wykonują mniej skomplikowany, okresowy ruch niezakłócony przez wykonywanie innych czynności. Ludzie poruszający się w miejscach publicznych bardzo często niosą różne przedmioty lub gestykulują, przez co bardzo trudno jest wybrać bazową, charakterystyczną, sekwencję ruchu dla każdej z przebadanych osób.

Biorąc pod uwagę powyższe warunki przedmiotem przeprowadzonych analiz był ruch kończyn dolnych w szczególności uda i podudzia. Ruch ten reprezentowany jest przez zestaw trójelementowych rotacji poszczególnych części ciała. W niektórych przypadkach przeprowadzonych badań dostosowano dane z komercyjnego systemu *MC* do informacji z systemów wideo poprzez redukcję liczby analizowanych sygnałów rotacji do rotacji względem osi *X*. Tak zredukowane dane symulują informacje dostarczone z systemów wideo, które są częściej używane do zastosowań związanych z bezpieczeństwem. W ten sposób możliwe będzie użycie podobnych algorytmów w już istniejących systemach.

### 1.4 Prowadzone badania

Szerokie spektrum zastosowań technologii tworzenia modeli ruchu w celu identyfikacji osób uwidacznia się w różnorodności rozwijanych projektów. W USA, po wydarzeniach zaistniałych 11 września 2001r, duży nacisk kładzie się na implementację tych technologii w projektach związanych z bezpieczeństwem. Jednym z nich jest projekt „Human ID at a distance" finansowany przez *DARP* - jednostkę Departamentu Obrony USA. Uczestniczące w nim zespoły m.in. z ośrodków Carnegie Mellon, Georgia Tech, MIT realizują projekt automatycznego rozpoznawania chodu w celu identyfikacji postaci ludzkiej na odległość. Grupa Metaxasa [8] z Rutgers University, NJ; zajmuje wykorzystaniem podobnych technik w zastosowaniach medycznych.

W Azji intensywnie rozwija się projekt *CBAT* (*Center for Biometric Authentication and Testing*) mający na celu wykorzystanie technik biometrycznych w celu zwiększenia bezpieczeństwa narodowego Chin, a także konkurencyjności Chińskiej nauki na kreującym się właśnie rynku biometrycznych technologii

W Europie dokonuje się również wielu prób wykorzystania technik modelowania kształtu i ruchu postaci ludzkich do zastosowań medycznych. Mają one na celu rozpoznawanie wad postawy, zniekształceń powstałych w wyniku urazów oraz przyspieszania procesu rehabilitacji. Jedną z takich inicjatyw jest projekt *CAREN* [17] (*Computer Assisted Rehabilitation Environments*) rozwijany w Holandii przy udziale środowisk naukowych (Akademia Medyczna) i firm informatycznych (Silicon Graphics, IBM).

Rozbudowany system identyfikacji jest opracowywany przez grupę z Southampton University [2, 3, 9, 14, 15]. Prezentowane prace wskazują na duży stopień trafnych identyfikacji. Poza biometrycznym wykorzystaniem ruchu grupa ta zajmuje się także innymi technikami i metodami biometrycznymi m. in. identyfikacją przez porównywanie kształtu małżowiny usznej.

Zanchi [13] prowadzi intensywne badania nad wykorzystaniem ruchu w biometrii. Zajmuje się m. in. trójwymiarową analizą ruchu człowieka w oparciu o obraz zarejestrowany przy pomocy zsynchronizowanych kamer wideo. W obrębie jego prac znajduje się również badanie możliwości zastosowania technik biometrycznych opartych o analizę ruchu w aspektach związanych z medycyną.

W Polsce technikami biometrycznymi zajmuje się między innymi Laboratorium Biometrii Politechniki Warszawskiej. Koncentruje się ono jednak głównie nad identyfikacją człowieka na podstawie wzoru siatkówki oka bądź na podstawie odcisków palców. Opracowaniem modeli ruchu zajmują się takie ośrodki jak Centrum Zdrowia Dziecka w Międzylesiu, Centralny Instytut Ochrony Pracy, Szpital Kliniczny w Zagórzu [4]. Rozległy zakres tematyki oraz stopień trudności problemu pozostawia wiele miejsca dla pracy kolejnych zespołów badawczych z innych instytucji rządowych i pozarządowych.

## 2. Widmo amplitudowe

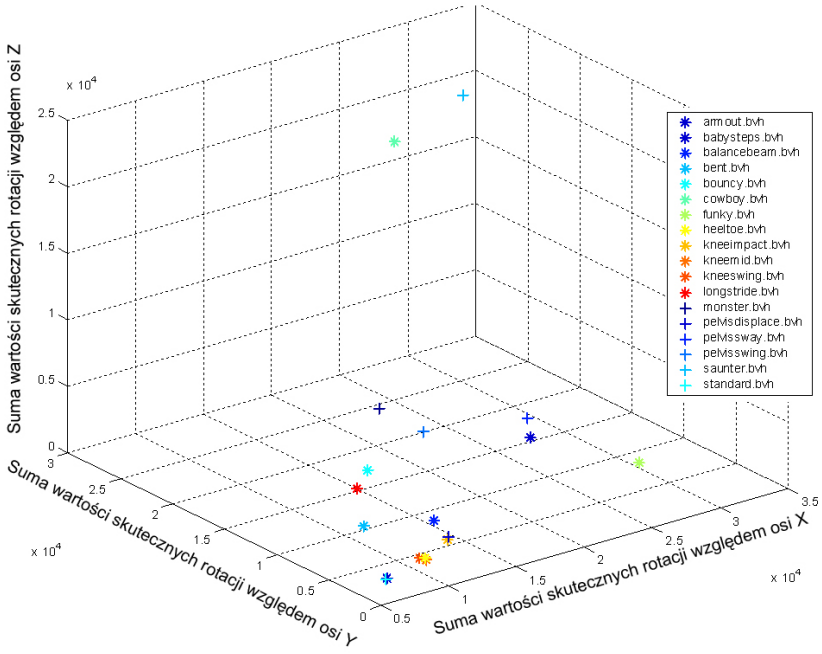
Na podstawie cytowanych prac [2, 3, 9, 14, 15] przypuszcza się, że analiza widma amplitudowego sygnału ruchu kończyn dolnych może dawać obiecujące wyniki. Przeprowadzone badania miały na celu zweryfikowanie tego twierdzenia na potrzeby analizy ruchu uzyskanego z komercyjnego systemu *MC*. Widmo amplitudowe sygnałów rotacji powstaje przez poddanie ich działaniu dyskretnej transformacji Fouriera wyrażonej wzorem:

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\,\pi i k n / N} \tag{0}$$

Uzyskane w ten sposób widmo podlega dalszej analizie. Widmo amplitudowe tak jak widmo fazowe jest źródłem cennych biometrycznych informacji. Kuan [5] pokazał, że parametry widma amplitudowego poszczególnych osób znacząco różnią się o siebie. W pracach [14, 15] podejście to zostało rozszerzone przez zastosowanie widma amplitudowo fazowego (ang. *phase-weighted magnitude*). W niniejszej pracy użyto analogicznego sposobu reprezentacji wyników, aby możliwe było ich porównywanie.

### 2.1 Wartości skuteczne harmonicznych

Na potrzebę przeprowadzonych badań, przyjęto uwzględnienie pierwszych 30 wartości skutecznych harmonicznych. Poza tym pasmem zgromadzona jest bardzo niewielka część energii. Najprostszym sposobem jest porównanie energii zgromadzonej w tym zakresie widma dla sygnałów rotacji uda w płaszczyznach *X*, *Y* i *Z*. Na rysunku 1 przedstawiony jest wykres będący wynikiem tak skonstruowanej analizy. Na osiach X, Y i Z odpowiednio znajduje się suma wartości trzydziestu pierwszych harmonicznych sygnałów rotacji uda względem osi *X*, *Y* i *Z*. Poszczególne punkty reprezentują różne sekwencje ruchów wymienione w legendzie.




**Rys. 1.** Sumy wartości skutecznych 30 harmonicznych sygnałów rotacji odpowiadającym poszczególnym osiom *X, Y, Z*

### 2.2 Wartości pierwszych znaczących harmonicznych

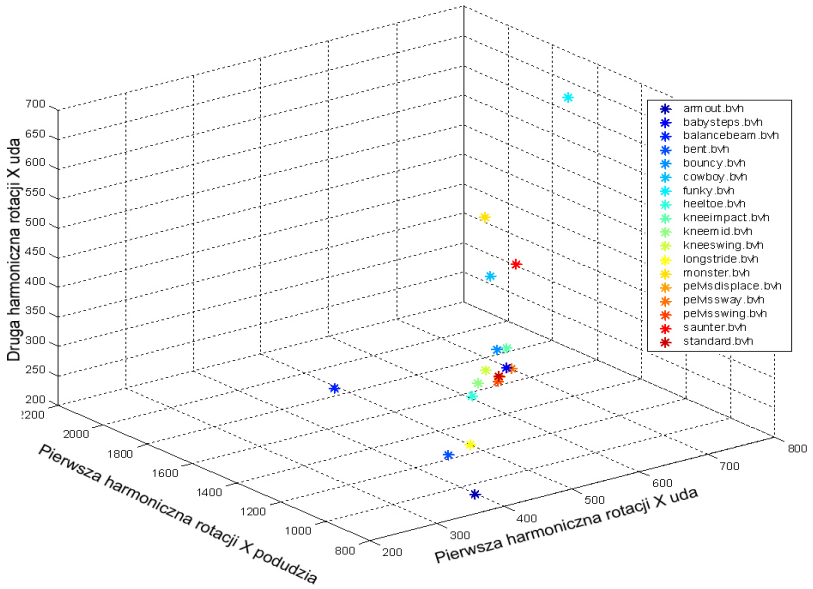
Wadą analizy bazującej na sumie wartości skutecznej wybranej liczby harmonicznych jest fakt, iż operacja sumowania uśrednia w pewien sposób pomiary, co powoduje utratę znaczących informacji. Aby pominąć tą niedogodność, ograniczono się to analizy widma amplitudowego tylko z okresów wybranych sygnałów rotacji. Okazuje się, że dla prezentowanych sygnałów najczęściej pierwsze dwie składowe harmoniczne niosą znaczną ilość informacji. W tym doświadczeniu analizie poddane zostały sygnały rotacji *X* uda i podudzia. Pod uwagę brane były wartości pierwszego i drugiego prążka rotacji uda i pierwszego prążka rotacji podudzia. Jest to zastosowanie podejścia przedstawionego w pracy [13]. W tym przypadku jednak analiza dotyczy sygnałów systemu *MC* zamiast sekwencji wideo. Wyniki tak przeprowadzonego eksperymentu przedstawione są na rysunku 2.

## 3. Metoda dynamicznej transformacji dziedziny czasu

Metoda *Dynamic Time Warping* (*DTW*) jest powszechnie wykorzystywana do porównywaniu ze sobą sygnałów różnej długości (mowy [9], także rozpoznawania pisma). Jej zastosowanie do ruchów przedstawione zostało w pracy [6] w kontekście porównywania sekwencji ruchów. Metoda może być użyta także do łączenia ruchów podstawowych w dłuższe, bardziej skomplikowane sekwencje [7]. Próba organizacji bazy danych ruchów przedstawiona została w pracy [1].

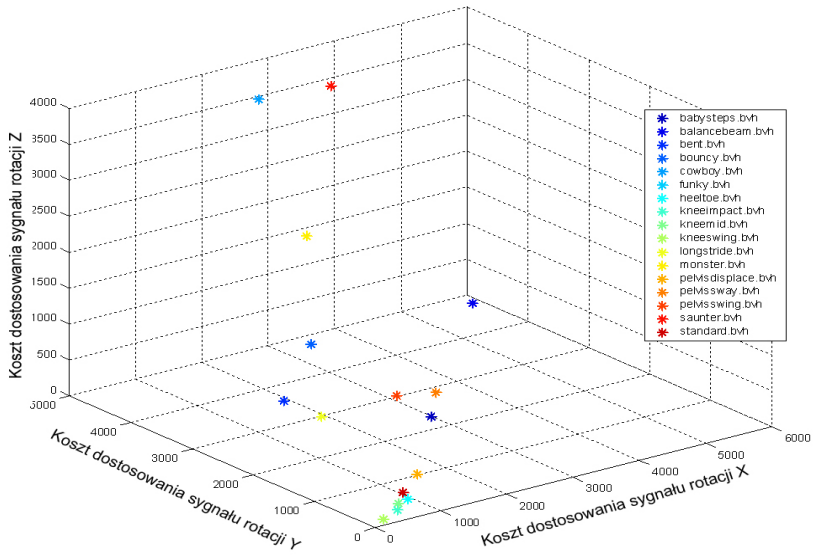
### 3.1 Sygnały rotacji w trzech wymiarach

Badania oparte na *DTW* mają odmienną naturę od tych bazujących na analizie widma amplitudowego ruchu człowieka. Te drugie mogą stanowić podstawę wielokryterialnego procesu klasteryzacji ruchów, która będzie miała na celu stworzenie biometrycznej bazy danych ruchów ludzkich. Metoda *DTW* służyć może bezpośrednio do porównywania danych sekwencji ruchu w celu oceny ich podobieństwa. W metodach bazujących na widmie amplitudowym to zadanie spełniać może metoda k– najbliższych sąsiadów opisana poniżej.




**Rys. 2.** Analiza własności widma amplitudowego rotacji uda i podudzia.

W pierwszym doświadczeniu, zgodnie z przedstawionym wcześniej opisem, przy pomocy metody *DTW* porównywano sygnały rotacji *X*, *Y* oraz *Z* uda. Materiał badawczy pochodził z grona wyselekcjonowanych plików *MC*. Ich wspólną cechą było przechowywanie informacji o ruchu przeprowadzonym w kierunku obserwatora. Ruch ten odbywał się w linii prostej. Spośród tych plików został wybrany jeden wzorcowy (*armout.bvh*) reprezentujący najbardziej typowy ruch, do którego badane było podobieństwo pozostałych. Każdy sygnał różnił się od pozostałych m.in długością. Zgodnie z własnościami metody *DTW* długości wszystkich badanych sygnałów były dostosowywane do długości sygnału wzorca. Koszt niniejszej operacji jest miarą podobieństwa sygnałów. Koszt dostosowania sygnałów względem odpowiednich osi stanowi współrzędne punktów w przestrzeni podobieństwa sygnałów. Tak uzyskane wyniki widnieją na rysunku 3.

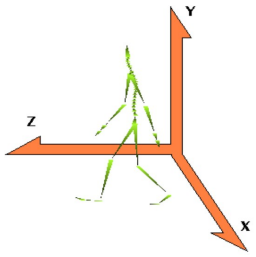



**Rys. 3.** Koszt dostosowania sygnałów rotacji osi *X, Y, Z* metodą *DTW* do sygnału *armout.bvh*

Otrzymane w taki sposób wykresy należy interpretować inaczej niż w przypadku metod opartych na analizie widma amplitudowego – punkty bliższe początkowi układu współrzędnych reprezentują ruchy bardziej podobne do wzorca.

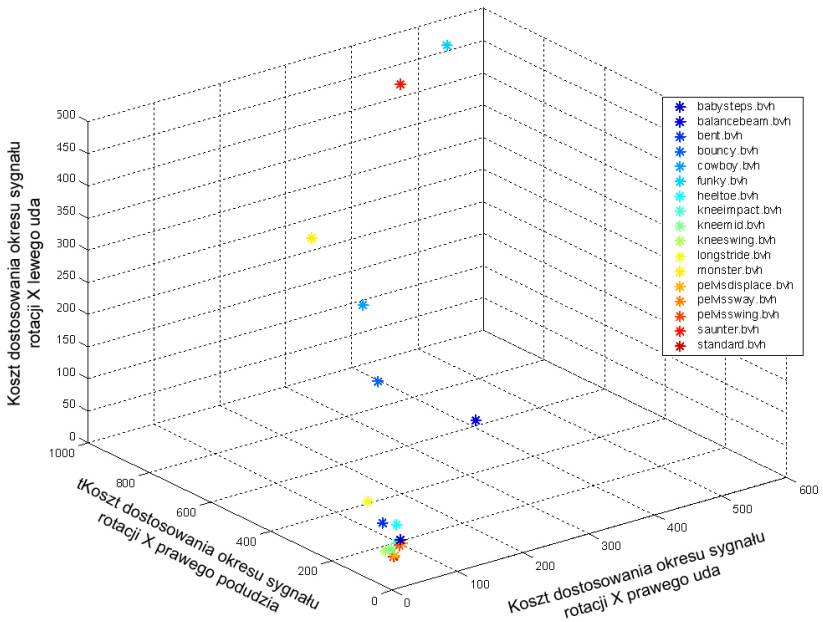
### 3. 2 Sygnały rotacji w jednym wymiarze

Systemy *MC* pomimo tego, iż są bardzo dokładne, mają bardzo ograniczone możliwości zastosowań. W praktyce o wiele częściej spotykane są systemy rejestracji wideo. Aby zasymulować dane dostarczane z takiego systemu na potrzeby niniejszego eksperymentu, ograniczono się do rotacji względem osi X. Jest to równoznaczne z analizą materiału przy ruchu prostopadłym do kamery (rysunek 4).




**Rys. 4.** Układ współrzędnych w badanych plikach na przykładzie standard.bvh.

W badaniach brano rotacje X prawego uda, podudzia oraz lewego uda. Naturalnym podejściem byłoby zbadanie ruchu stopy, co jednak jest trudne do przeprowadzenia w praktyce. Zaletą tak skonstruowanych parametrów ruchu jest także wrażliwość na niesymetryczności pomiędzy ruchem lewego i prawego uda. Wyniki analizy przedstawia rysunek 5. W tym przypadku widać wyniki dostosowania okresów wyżej wymienionych sygnałów rotacji. Badanie części z zarejestrowanego sygnału występuje w sytuacji w praktyce, gdy sygnał jest znacznie zakłócony i niedostępny w całości do rejestracji.




**Rys. 5.** Koszt dostosowania okresów sygnałów rotacji *X* (prawe udo, prawe kolano, lewe udo) metodą DTW do analogicznych okresów sygnałów sygnału wzorcowego *armout.bvh*.

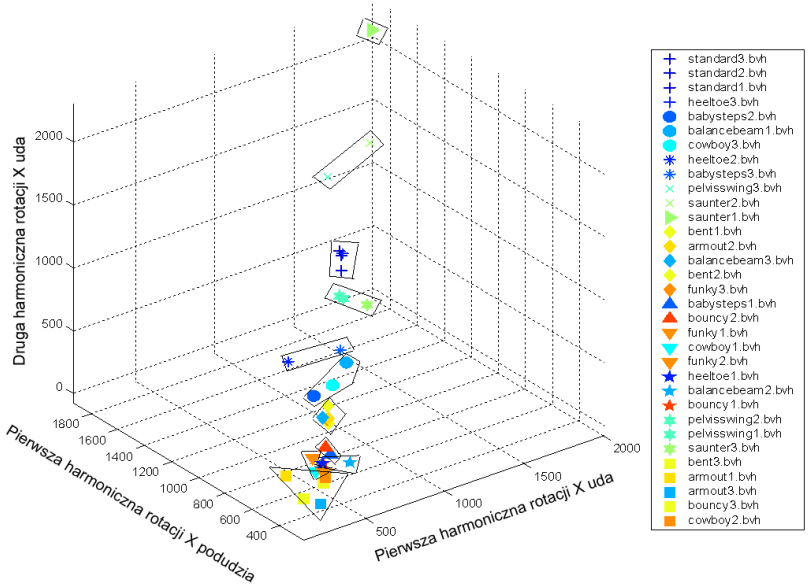
## 4. Klasteryzacja

Proponuje się przeprowadzić klasteryzację badanych sekwencji ruchów. Jej celem ma być określenie, czy ruchy pochodzą od jednego osobnika, czy też znacznie różnią się od siebie. Konstruowany proces będzie rodzajem klasyfikacji „bez nadzorcy”. Dalszym krokiem może być opracowanie właściwych metod identyfikacji nowych danych przez klasteryzację „z nadzorcą”. W takim przypadku z góry określone są grupy ruchów, do których

porównywana jest nowa sekwencja i określane jest podobieństwo, a przez to przynależność ruchu do danej klasy. W prowadzonych badaniach zostanie użyta metoda *k-means*.

Najprostsza wersja metody *k-means* polega na wybraniu *k* obiektów z analizowanej grupy, które będą spełniać rolę środków klastrów. Następnie każdy element jest przyporządkowywany do najbliższego klastra (do najbliższego środka klastra przy zdefiniowanej metryce). Po przyporządkowaniu elementów wyznaczany jest nowy środek ciężkości klastrów. Kolejne iteracje są wykonywane tak długo, aż środki klastrów przestają się poruszać (lub zmieniają się poniżej pewnej wartości granicznej). Główną wadą tego typu algorytmu jest konieczność określenia liczby klastrów przed rozpoczęciem procesu.

Metoda ta jest powszechnie używana do różnych zastosowań. Na rysunku 6 przedstawiono wynik działania algorytmu *k-means* dla poprzednio rozpatrywanej grupy ruchów i analizy zaprezentowanej w rozdziale 2.2.




**Rys. 6.** Wynik działania algorytmu *k*-means użytego do klasteryzacji danych reprezentujących widmo amplitudowe sygnałów rotacji X uda i podudzia.

Większość klastrów jest dobrze odseparowana. Najlepsza sytuacja z punktu widzenia analizy biometrycznej występuje, gdy wszystkie różne sekwencje zostają rozdzielone pomiędzy różne klastry. Częściowo sytuacja taka występuje w analizowanym przypadku. Podstawowym problemem, jaki napotyka się w przedstawionych badaniach są słabo opisane dane wejściowe. W komercyjnej bazie ruchów, z której korzystano w badaniach, nie jest zaznaczone, czy ruchy te są wykonywane przez tego samego aktora. Analiza wyników klasteryzacji sugeruje, że jest pewne prawdopodobieństwo, że większość z badanych ruchów wykonywał jeden aktor. Aby potwierdzić tą tezę należałoby przeprowadzić bardziej szczegółowe badania przy użyciu dobrze wyposażonego laboratorium *Motion Capture*.

Proponuje się również rozwinięcie powyższej metody w celu przeprowadzenia wielowymiarowej analizy. Analiza przedstawiona w poprzednich rozdziałach pozwala na dobrą wizualizację w postaci trójwymiarowego wykresu. Można ją jednak rozszerzyć biorąc pod uwagę większą liczbą harmonicznych dla dwóch lub więcej stawów charakteryzujących ruch. W takim przypadku wymiarowość problemu znacznie wzrasta, jednak taka parametryzacja pozwoli na dokładniejsze określenie modelu ruchu charakteryzującego każdą osobę. Prowadzone są dalsze badania dotyczące tego rodzaju analizy.

## 5. Wnioski

W pracy omówiono najważniejsze podejścia mające na celu sprawdzenie hipotez dotyczących możliwości wykorzystania niektórych parametrów ruchu w celu identyfikacji osób. Jednym z nich było badanie widma amplitudowego, w szczególności sum wartości skutecznych pierwszych 30 harmonicznych widma rotacji *X*, *Y* i *Z* prawego uda. Powyższa metoda posiada jednak pewne niedogodności: uwzględnienie tylko wybranej części ciała, konieczność rejestracji sygnału na wszystkich trzech osiach oraz utrata informacji przy sumowaniu energii.

Wady powyższej metody skłaniają do podejścia przy biometrycznej analizie ruchu w podobny sposób, jak to zostało pokazane w pracach [2, 3, 9, 14, 15]. Bazuje się na rotacji X uda i podudzia, przez co dane *MC* zostają upodobnione do danych z sekwencji wideo. Jedną z zalet tego podejścia jest możliwość rozwinięcia wielokryterialnej klasteryzacji mogącej być podstawą projektowanego systemu identyfikacji człowieka przez analizę ruchu.

Dane poddawane analizie pochodzą z komercyjnego systemu *Motion Capture* [16]. Wadą materiału badawczego jest brak informacji na temat osoby wykonującej ruch zapisany w badanych sekwencjach. Nie można przez to jednoznacznie zweryfikować poprawności działania zaimplementowanych metod. Posiadanie takiego typu danych biometrycznych wymusiło skupienie prowadzonych badań bardziej na analizie podobieństwa danych sekwencji ruch niż na jednoznacznej próbie przyporządkowania poszczególnych danych do konkretnych osób.

Metoda DTW porównywania dwóch ciągów czasowych o różnych długościach pozwala na ocenę ich podobieństwa poprzez analizę kosztu przeprowadzonej transformacji dziedziny czasu. Przedstawione zostały wyniki analizy podobieństw ruchu dla sygnałów rotacji w trzech osiach. Współrzędne każdego z punktów w przestrzeni wyników są wartościami kosztu dopasowania sygnałów bądź okresów sygnałów rotacji różnych segmentów ciała ludzkiego. Wszystkie sygnały porównywane były do sygnałów odniesienia wygenerowanych z pliku *armout.bvh*. W większości przypadków można odseparować grupę plików wykazującą pewne podobieństwo zarejestrowanych w nich ruchów. W śród nich najczęściej pojawiają się: *armout.bvh, standard.bvh, pelvisdisplace.bvh, kneeswing.bvh, heeltoe.bvh, kneemid.bvh.* Można postawić hipotezę, że podobieństwo danych sekwencji ruchu wskazuje na wykonywanie ich przez tą samą osobę. Przeprowadzone badania należy traktować jako badania wstępne wyznaczające kierunki dalszych eksperymentów oraz wskazujące grupę metod mających w perspektywie zastosowanie w analizie biometrycznej ruchu.

Aby jeszcze bardziej zwiększyć dokładność prezentowanych metod w kolejnym kroku można zaproponować łączoną analizę czasowo-częstotliwościową. W takim przypadku należy określić miarę podobieństwa w dziedzinie czasu (za pomocą miary opartej na metodzie *DTW*) oraz analogicznej miary podobieństwa w dziedzinie częstotliwości (opartej na analizie pierwszych harmonicznych). Metoda taka może dać interesujące wyniki prowadzące do lepszej klasteryzacji sekwencji.

Zadaniem prezentowanych badań było sprecyzowanie kierunków, na których warto się skupić by skutecznie móc realizować plan budowy w pełni sprawnego systemu identyfikacji człowieka na podstawie analizy cech charakterystycznych ruchu. Wyniki różnych prac prowadzonych na całym świecie oraz badań przeprowadzonych na potrzeby niniejszego opracowania wskazują jednoznacznie, iż metody te są jak najbardziej wartościowe z biometrycznego punktu widzenia. Mogą one stanowić doskonałą podstawę do zbudowania biometrycznych baz wiedzy, które zawierać będą informacje na temat specyfiki ruchu zarejestrowanych w nich osób. W celu opracowania takiej bazy mogą z powodzeniem posłużyć metody bazujące na widmie amplitudowym. Klasteryzacja opracowana na podstawie parametrów widma oraz zastosowanie algorytmów klasyfikacji z uczeniem wydaje się być obiecującym rozwiązaniem.

Opracowanie dobrze odseparowanych, spójnych klastrów ruchu dla każdej osoby, a następnie dla każdego z nich odpowiedniego ciągu uczącego jest warunkiem koniecznym zbudowania oraz poprawnego działania biometrycznego systemu analizy ruchu w celu identyfikacji osób. System taki będzie mógł pracować na danych z systemu *Motion Capture*, a także z systemu opartego na technologii wideo. Klastry grupujące ruch będą dzieliły go pod kątem różnych osób, ale także pod kątem rodzaju wykonywanego ruchu (np. chód i bieg), a w dalszych, bardziej zaawansowanych etapach pod kątem stanów emocjonalnych uwidaczniających się w ruchu. Wymaga to jednak zbudowania dobrze sparametryzowanego modelu ruchu człowieka.

## Literatura

1. Bąk A. *Organizacja bazy danych ruchów elementarnych animowanych postaci ludzkich,* Praca Magisterska, Politechnika Wrocławska, Wrocław (2001).
2. Cunado D., Nixon M. S., Carter J. N., *Automatic extraction and description of human gait models for recognition purposes*, Computer Vision and Image Understanding, **90**(1), (2003).
3. Cunado D., Nixon M. S., Carter J. N., *Using Gait as a Biometric, via Phase-Weighted Magnitude Spectra*, Proc. of 1st Int. Conf. on Audio- and Video-Based Biometric Person Authentication, (2003).
4. Jasińska-Choromańska D., Korzeniowski D. Diagnostyka chodu – wzorzec chodu a pacjenci z implantami, w Pomiary Automatyka Kontrola, vol. **9 bis**, (2005) .
5. Kuan E. L., *Investigating gait as a biometric*, Technical report, Departament of Electronics and Computer Science, University of Sothampton, (1995).
6. Kulbacki M., Jabłoński B., Klempous R., Segen J. *Learning from Examples and Comparing Models of Human Motion*, Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. **8**, No. 5 (2004).

7. Kulbacki M. *Podstawowe metody syntezy ruchu animowanych postaci ludzkich*, Praca Magisterska, Politechnika Wrocławska, Wydział Elektroniki, Wrocław, (2001).
8. Metaxas D. *Physics-Based Deformable Models: Applications to computer vision, graphics and medical imaging,*, Kluwer Academics, (1997).
9. Mowbray S. D, Nixon M. S, *Automatic Gait Recognition via Fourier Descriptors of Deformable Objects*, Proc. of Audio Visual Biometric Person Authentication, (2003).
10. Murray M. P., *Gait as a total pattern of movement*, Amer. J. Phys. Med. **46** (1) (1967).
11. Murray M. P., Drought A. B., Kory R. C., *Walking patterns of normal men*, J. Bone Joint Surg. **46-A** (2) (1996) 335-360.
12. Rabiner L. R., Juang B. *Fundamentals of speech recognition*, Englewood Cliffs, N.J, Prentice Hall, (1993).
13. Zanchi V., Supuk T. *Human Motion Identification*, Workshop on Signals and Systems in Human Motion, SoftCOM 2004, Split-Dubrovnik-Venice.
14. Yam C. Y., *Automated person recognition by walking and running via model-based approaches*, Pattern Recognition **37** (5), Elsevier Science, (2003).
15. Yam C. Y., Nixon M. S., Carter J. N. *Extended Model-Based Automatic Gait Recognition of Walking and Running*, In Proceedings of Proceedings of 3rd Int. Conf. on Audio- and Video-Based Biometric Person Authentication, AVBPA 2001, (2001) pp. 278-283.
16 http://www.credo-interactive.com
17 http://www.e-motek.com/medical/index.htm

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.131-138
ISBN 83-922646-0-6


# Praktyczne aspekty implementacji narzędzia klasy Business Intelligence

Wojciech Kosiba

Centralny Ośrodek Informatyki Górnictwa S.A.
40-065 Katowice
ul. Mikołowska 100
607-468-348
Wojciech.Kosiba@coig.katowice.pl

**Streszczenie:** W referacie omówiono możliwości funkcjonalne narzędzia klasy Business Intelligence na przykładzie rozwiązań wykonanych przez COIG S.A. dla klientów z branży wydobycia węgla kamiennego, w oparciu o technologię Business Objects i przy zastosowaniu specjalizowanej dla celów hurtowni danych bazy danych Sybase IQ. Przyjęto krokowy model wdrożenia, wprowadzając ograniczenia zarówno w sferze tematycznej jak i funkcjonalnej. Pierwszym etapem wdrożenia rozwiązania stał się obszar tematyczny Zbytu Węgla w zakresie informacji dostępnych na fakturze. Już na tym etapie stało się możliwe tworzenie takich zestawień, jakich wcześniej nie przewidywali autorzy założeń hurtowni danych i jej warstwy prezentacyjnej, stało się zatem możliwe obserwowanie niedostępnych dotąd przekrojów wielowymiarowej bazy danych. Kolejnymi obszarami dla których zbudowano Tematyczne hurtownie danych (ang. Data Mart, to Logistyka Materiałowa, Koszty oraz Kadry. Referat powstał w oparciu o doświadczenia zdobyte w ciągu wdrożeń systemów eksploatowanych dla celów biznesowych, stąd ściśle praktyczne spojrzenie na proces zaprezentowane przez autora referatu.

## 1. Wprowadzenie

Efektywne funkcjonowanie każdej firmy wymaga wsparcia decyzji kadry zarządzającej odpowiednio przygotowaną informacją na temat aktualnych parametrów ekonomicznych, lub takich których konsekwencje mogą rzutować na wyniki ekonomiczne. Nowatorskim aspektem rozwiązań przy użyciu narzędzi klasy Business Intelligence jest to, że do rąk użytkownika-analityka merytorycznego, trafia narzędzie umożliwiające eksplorację baz danych, dotychczas dostępną jedynie dla wyspecjalizowanych programistów bazodanowych [1]. Dotychczas osoba zajmująca się zagadnieniami analityczno-merytorycznymi zmuszona była teoretycznie sama wymyślić treść zestawienia, opracowania czy wykresu. Osoba ta nie mając dostępu do bazy danych, mogła obserwować jedynie dostępne jej preselekcjonowane informacje utworzone w formie wydruków, grafiki czy nawet w internetowej lub elektronicznej wersji sztywnych raportów. Obecnie po wdrożeniu rozwiązań klasy Business Intelligence osoba ta dostaje narzędzie o funkcjonalności języka zapytań SQL do bazy danych [2]. Moim zdaniem jest to rewolucyjna zmiana w podejściu do danych. Już nie programista, czy projektant systemu tworzy zestawienia, które mogłyby być użyteczne dla analityka merytorycznego, tylko sam analityk uzyskuje dostęp do wielowymiarowej kostki danych, którą może przecinać dowolnymi dwuwymiarowymi płaszczyznami lub nawet trójwymiarowymi przestrzeniami. Wielowymiarowość niegdyś postrzegana jako obiekt ściśle teoretycznych rozważań, uzyskuje realny, a nawet bardzo prosty kształt. Kolejną konsekwencją wdrożenia hurtowni danych wraz z jej warstwą prezentacyjną jest to, że użytkownik-analityk merytoryczny uzyskuje możliwość porównania danych z różnych, dotychczas rozdzielnych źródeł. W skrócie mówiąc, można posłużyć się przenośnią, że dotychczas osoba, do której obowiązków należało analizowanie informacji o parametrach firmy miała funkcjonalność typu pakietu Excel na danych takich, jakie udało się jej zdobyć. Dzisiaj osoba ta otrzymuje język zapytań SQL dostępu do baz danych, nie tracąc nic z możliwości modelowania danych do postaci tabeli, macierzy czy wykresów różnych typów.

Dosłownie tłumacząc angielski termin „Business Intelligence" otrzymujemy „wywiad gospodarczy", czyli jak najobszerniejszą informację o przedsięwzięciu biznesowym i jego otoczeniu. Istotą hurtowni danych jest możliwość wprowadzenia wszelkich posiadanych informacji i zorganizowania ich w sposób adekwatny do możliwości zastosowanej bazy danych. Z drugiej zaś strony, inteligencja to umiejętność kojarzenia faktów pozornie odległych. Dobrze zorganizowana hurtownia danych daje możliwość kojarzenia wymiarów dotychczas niekojarzonych. Przykładowo, bardzo proste jest stworzenie zestawienia w wymiarze geograficznym sprzedaży

węgla według kaloryczności i sortymentu. Dzisiaj takie zestawienie można stworzyć i natychmiast z niego zrezygnować. Można też wyciągnąć wnioski i zbudować inne zestawienie, w którym wykorzystane zostaną uzyskane wcześniej informacje. Przedtem uzyskanie takiego zestawienia wymagało zatrudnienia programisty znającego narzędzie SQL, co było kosztowne, a jego potrzeba wątpliwa.

Budując hurtownię danych należy brać pod uwagę możliwość wprowadzenia w przyszłości do niej takich informacji, które dzisiaj w ogóle nie są zbierane. Mogą to być np. informacje o trendach sprzedaży u konkurencji, uzyskane zupełnie poza systemami informatycznymi eksploatowanymi w firmie, np. kupione w profesjonalnej wywiadowni gospodarczej.

Wdrożone aplikacje wymagają ciągłej modernizacji pod kątem zarówno informacji, które uzyskiwane są na wyjściu z systemu, jak i architektura hurtowni danych musi uwzględniać zmieniającą się strukturę systemów źródłowych.

Ważnym i obecnie niedocenianym elementem pośród czynności, które należy wykonać wokół wdrożenia hurtowni danych i rozwiązań Business Intelligence jest, jak to już wcześniej podkreślono, wsparcie analityka merytorycznego w zakresie eksploracji danych. Bezpośrednio po wdrożeniu analityk merytoryczny z jednej strony zaczyna zauważać aspekty dotychczas niewidoczne, czyli takie zestawienia, których istnienia nie podejrzewał, z drugiej zaś strony styka się z danymi, do których wcześniej dostęp mieli jedynie programiści bazodanowi. Bardzo istotne jest, aby na tym etapie panować nad wymiarami, które zostaną zaimplementowane w Świecie Obiektów, gdyż może się zdarzyć, że dane miały dotychczas charakter „techniczny". Udział autorów systemu źródłowego w budowie ETL może skrócić czas implementacji nawet o 75%.

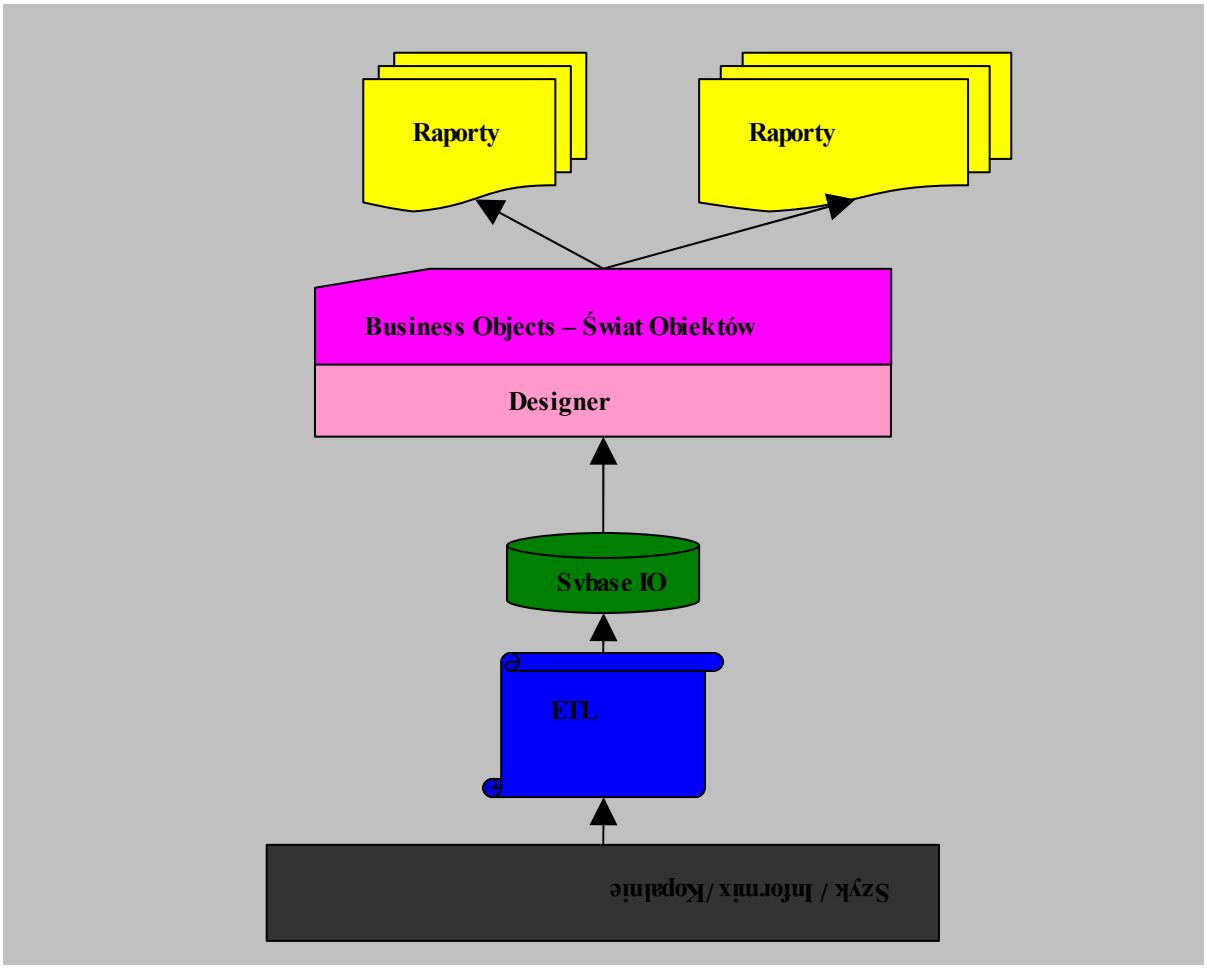
System Business Intelligence może być tak skonfigurowany, aby można było o dowolnej porze uzyskać operacyjną informację na temat najważniejszych, interesujących nas parametrów ekonomicznych. Taki dostęp zwany „kokpitem menadżerskim" umożliwia obserwowanie dynamicznych zmian i relacji pomiędzy nimi za pomocą np. graficznych emulatorów wskaźników analogowych.

## 2. System źródłowy SZYK/INFORMIX

Istotnym czynnikiem efektywnego wdrożenia hurtowni danych jest kontakt z twórcami systemu źródłowego. W przypadku wdrożenia hurtowni danych u klientów COIG S.A. aspekt ten nie był krytyczny, ponieważ autorem systemu źródłowego Zintegrowanego Systemu Wspomagającego Zarządzanie Przedsiębiorstwem – System SZYK, jest również COIG S. A. [3, 4]. Programowanie hurtowni danych wykonywane było w specjalistycznej pracowni, co powodowało konieczność ścisłych kontaktów również z innymi pracowniami systemu SZYK zajmującymi się rozwiązaniami dla poszczególnych obszarów tematycznych. Pracami objęto: obszar Zbytu Węgla, obszar Logistyki Materiałowej, obszar Rozliczenia Kosztów, obszar Ewidencji Kadrowej oraz obszar Centralnych Rozrachunków. Istota współpracy z autorami systemu źródłowego polega na wyselekcjonowaniu takich danych z Baz Tematycznych Systemu Szyk, które mają wartość merytoryczną. W toku kilkuletniego budowania baz danych dla klientów, niektóre pola mogły zmienić swoją funkcjonalność, inne zaś przestały być wykorzystywane i zawierają nieistotne z merytorycznego punktu widzenia informacje lub w skrajnych przypadkach wartości przypadkowe.

Można sobie wyobrazić wdrożenie narzędzi klasy Business Intelligence prowadzone na nieautoryzowanych danych, wówczas szacuje się wydłużenie procesu wstępnego projektu o 100%. Wykorzystywana przez system SZYK relacyjna baza danych Informix jest bardzo praktycznym narzędziem informatycznym wspierającym aplikacje transakcyjne. Jednak próba wykorzystania narzędzia Informix bezpośrednio jako warstwy bazodanowej hurtowni danych nie powiodła się. Komputer typu IBM F80 z systemem operacyjnym AIX częściowo obciążony pracą na rzecz aplikacji transakcyjnych ulegał permanentnym przeciążeniom. Dlatego też podjęto decyzję o uruchomieniu specjalizowanej bazy danych Sybase IQ, która jest zorganizowana w ten sposób, że dane przechowywane są jako poindeksowane kolumny. W celu przejścia do efektywniejszego środowiska bazodanowego, odrzucenia informacji zbędnych i szkodliwych, zapewnienia właściwej kontroli danych, sformatowania danych do potrzeb nowego środowiska - zastosowano technikę zbierania danych ETL.

## 3. Technika zbierania danych - warstwa ETL




**Rys. 1** Schemat ideowy rozwiązania

Zastosowana w rozwiązaniach wdrożonych u klientów COIG S.A. technika zbierania danych – warstwa ETL (ang. extract, transformation, load) oznacza ekstrakcję z systemu źródłowego oraz transformację danych ze względu na potrzeby, o których wspomniano pod koniec poprzedniego punktu referatu oraz ładowanie danych do systemu docelowego. Każdy z obszarów tematycznych hurtowni danych (ang. Data Mart) charakteryzował się innymi potrzebami. I tak np. w obszarze Zbytu Węgla w rozwiązaniach przygotowanych na bazie Informix wykorzystywano zbyt wiele tablic, co w początkowym okresie nastręczało olbrzymich trudności w wychwyceniu i zaimplementowaniu powiązań pomiędzy danymi w ten sposób, aby na etapie tworzenia raportów, czyli dwuwymiarowych przekrojów bazy danych otrzymywać dane spójne logicznie oraz zgodne formalnie. Oczywistym jest, że narzędzie Business Objects jako generator zapytań w języku SQL może generować bardzo długie zapytania do bazy danych, jednak zapytania te zawsze muszą być zbudowane w sposób logiczny i formalnie zgodny nie tylko ze składnią języka SQL, co jest wbudowane w system, ale zgodny z logiką danych opisanych przez programistę podczas tworzenia Świata Obiektów[5],[6]. W niektórych sytuacjach wygodniej i efektywniej z punktu widzenia architektury narzędzia Business Intelligence jest przygotować dane w taki sposób, aby relacje opisane w bazie danych były maksymalnie uproszczone. W obszarze tematycznym Zbytu Węgla zdecydowano się na utworzenie w warstwie ETL jednej tablicy zawierającej wszystkie informacje z faktury. Ta jedna tablica umożliwia nam tworzenie przekrojów sprzedaży węgla zarówno np. w wymiarze ilości i ceny węgla jak i w wymiarze parametrów jakościowych, czy chemicznych. Pozostałe tablice związane z wymiarem czasu, wymiarem klienta pozostawiliśmy poza tablicą danych podstawowych.

## 4. Specjalistyczna baza danych – Sybase IQ

Zastosowana do zbudowania hurtowni danych baza danych Sybase IQ charakteryzuje się pionowym składowaniem tablic oraz tym, że tablice są logiczne, a nie fizyczne. Daje to efekt 90% redukcji operacji wejścia wyjścia, przy czym przetwarzane są tylko kolumny niezbędne do zapytania. Drugą bardzo charakterystyczną cechą bazy danych Sybase IQ jest to, że dane tworzą indeks. Jest to opatentowana metoda redukcji danych do bitów „Bit Wise". Występują indeksy bitmapowe, indeksy do dat, czasu, słów kluczowych oraz indeksy do złączeń. W wyniku zastosowania tych metod nigdy nie występuje skanowanie tablic, a zapytania „ad hoc" działają tak jakby baza była strojona do każdego z nich. Zazwyczaj sumaryczny rozmiar bazy danych jest mniejszy niż czyste dane źródłowe.

Indeks bitmapowy zakładany jest przez użytkownika, gdy liczba unikalnych wartości w kolumnach zawiera się pomiędzy 10 a 500. Indeks „Bit-Wise" zakładany jest przez użytkownika, gdy ilość wystąpień przekracza 1000. Nie należy wówczas używać poleceń typu „distinct", W przypadku gdy chcielibyśmy używać „distinct" należy zastosować indeks „Bit-Wise"+B-drzewo.

Mówiąc językiem praktyki zastosowanie bazy danych Sybase IQ ułatwiło wdrożenie hurtowni danych oraz przyspieszyło proces rozwoju hurtowni wraz z narzędziem klasy Business Intelligence.

## 5. Opis danych – Designer

Narzędziem umożliwiającym tworzenie Świata Obiektów dla Business Objects jest oprogramowanie pozwalające opisać dane, nadać im łatwe do zrozumienia dla analityka merytorycznego nazwy oraz opisać relacje pomiędzy danymi – narzędzie Designer. Narzędzie to jest bardzo intuicyjne, a jego używanie dla doświadczonego programisty języka zapytań SQL jest wręcz oczywiste. Graficzny interfejs tego narzędzia pozwala posługiwać się metodą „przeciągnij i upuść".

## 6. Warstwa – Świat Obiektów

Utworzony za pomocą narzędzia Designer Świat Obiektów jest najważniejszym elementem projektu hurtowni danych. W skrócie mówiąc warstwa Świat Obiektów w prosty sposób realizuje wizję danych w oczach merytorycznego specjalisty. O ile w bazie danych nazwy niektórych pól są bardzo skomplikowane i nazwane w jakiś mnemotechniczny sposób na użytek programistów dane te otrzymują nazwy spójne i zrozumiałe. Na przykład, jeśli pole „f100" w bazie danych oznacza tak naprawdę numer faktury, to w warstwie Świat Obiektów wskazane jest aby wymiar ten nazywał się po prostu „numer faktury". Należy podkreślić, że implementacja narzędzia Business Objects nie wymaga żadnych zmian w strukturze wykorzystywanych baz danych, wszystkie transformacje i nowe powiązania tworzone są albo w warstwie ETL albo w warstwie Świata Obiektów. Należy również podkreślić, że nawet jeśli tablice w źródłowej bazie danych duplikują informacje bądź zawierają je w różnych miejscach nie ma to wpływu na opis w warstwie Świata Obiektów. Świat Obiektów może być zorganizowany w sposób spójny i logiczny merytorycznie

## 7. Dynamiczne Tworzenie Raportów

Absolutnym novum wynikającym ze stosowania narzędzia klasy Business Intelligence jest możliwość dynamicznego zmieniania i tworzenia nowych raportów. W warstwie prezentacyjnej prócz funkcji pozwalających np. na automatyczne wstawienie wykresu, czy dowolnego przekształcenia tabeli z danymi, istnieje możliwość stosowania rozbudowanego aparatu matematycznego. Dostępna jest większość funkcji matematycznych umożliwiająca prowadzenie analiz ekonomicznych. W skrócie mówiąc, w rękach zaawansowanego analityka biznesowego znajduje się cały aparat dostępny programiście i tylko kwestią strojenia wydajnościowego pozostaje, w której warstwie zaimplementować wymagane obliczenia. Należy zaznaczyć, że w przypadku bardzo często wykorzystywanych danych warto stworzyć odpowiedni zapis w bazie danych, na której oparto hurtownię danych, wówczas konieczne może okazać się rozbudowanie procedur warstwy ETL. W przypadku rzadziej wykorzystywanego przekroju danych można stworzyć prekalkulowany obiekt w Świecie Obiektów Business Objects. Ewentualnie jeżeli dana pozycja jest w fazie testowej, jej wyliczanie można pozostawić w warstwie tworzenia raportów. Jeżeli pozycja miałaby być wykorzystywana w dalszej fazie analiz, można ją dodatkowo zapisać w raporcie jako obiekt <wymiar, szczegół lub miarę>. Funkcjonalność narzędzia Business Objects umożliwia pracę zbiorową, ponieważ raport zapisywany jest wraz z tworzącym go zapytaniem. Po przesłaniu innemu analitykowi raport może powrócić do fazy tworzenia zapytania.

## 8. Organizacja Pracy – Czyli Co dla kogo ?

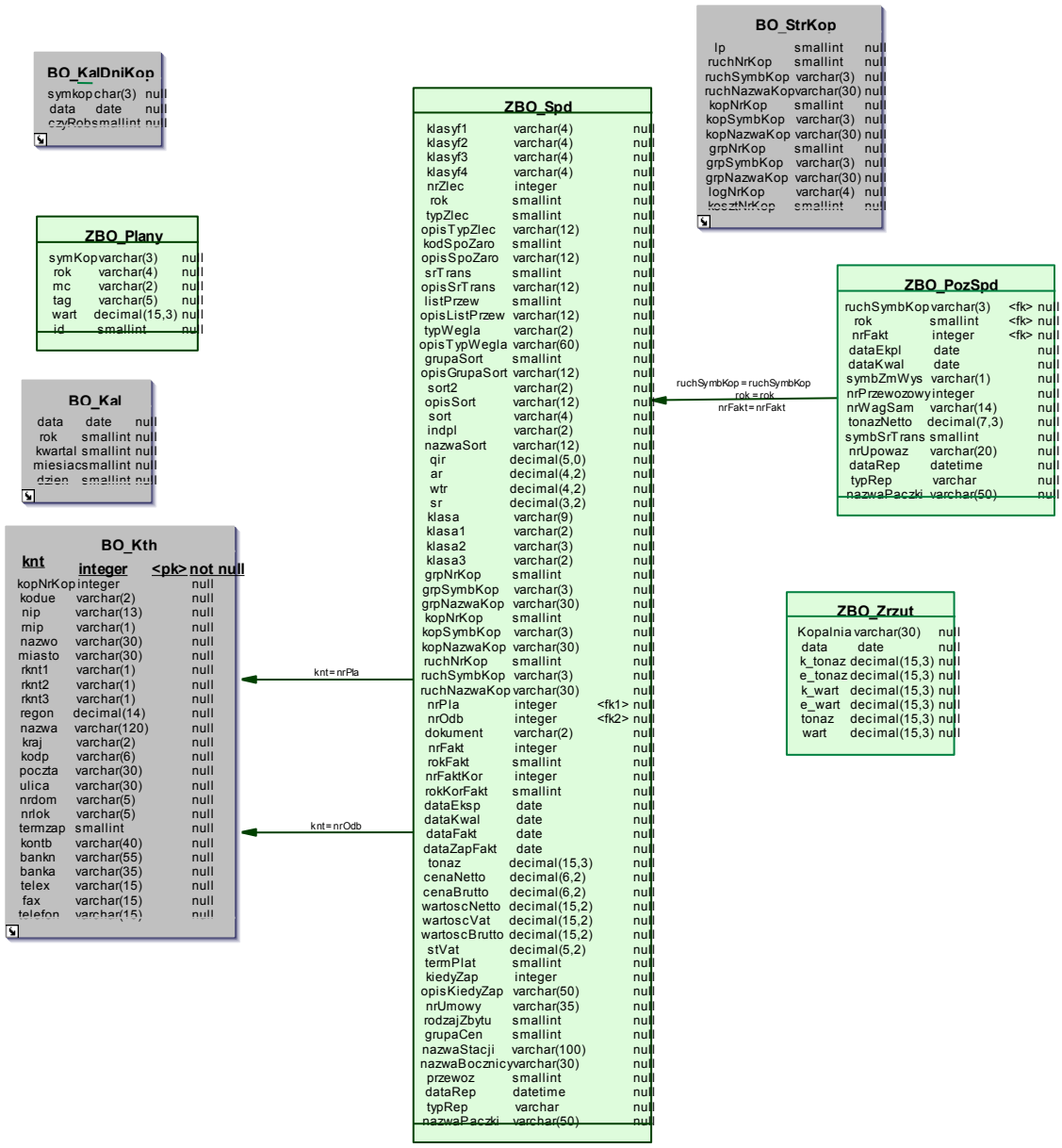
Na początku tego punktu przytoczę cytat „Pierwsze zapytanie marketingowe, nigdy nie jest właściwym. To jest proces interakcyjny : zapytanie, odpowiedź, zapytanie, aż do wniknięcia w sens biznesu" – Pete Ester[5]. System Business Intelligence jest systemem, który umożliwia tworzenie raportów. Wydawałoby się więc, że największe zastosowanie znajdzie on dla pracowników związanych z szeroko rozumianą sprawozdawczością. Takie wykorzystanie oczywiście jest możliwe, za pomocą Business Objects można generować np. zestawienia sprawozdawczości obowiązkowej. Wydaje się jednak, że bardziej efektywne bardziej naturalne jest udostępnienie Business Objects pracownikom komórek zajmujących się Informacją Zarządczą tj. szybką informacją, umożliwiającą sprawne podejmowanie operacyjnych decyzji z przedsiębiorstwie. Generalnie w systemie Business Intelligence rozróżnić należy następujące role :

- **Końcowy odbiorca raportów** – może to być zarówno manager wysokiego szczebla jak i na przykład kierownik magazynu. Ten pierwszy otrzymywałby informacje strategiczne, drugi zaś np. szczegółowe zestawienie stanów na określonych grupach materiałowych.
- **Analityk merytoryczny** – posiadający dostęp do Świata Obiektów i do edycji innych raportów, osoba taka odpowiedzialna byłaby za tworzenie raportów dla poprzedniej kategorii pracowników.
- **Designer** – osoba po stronie twórców oprogramowania, odpowiedzialna za stworzenie takiego Świata Obiektów, który będzie łatwy do zrozumienia dla grupy analityków merytorycznych.

## 9. Data Mart – Zbyt Węgla

Baza danych systemu Zbyt Węgla składa się z jednej tablicy zasadniczej i 7 tablic pomocniczych.

Na podstawie powyższego schematu widać różnicę w organizacji bazy danych przygotowanej dla hurtowni danych w stosunku do standardowej bazy relacyjnej. W celu pełniejszego zobrazowania budowy hurtowni zaprezentuję jeszcze zawartość głównej tablicy faktur.




**Rys. 2** Tablice systemu Zbyt Węgla.

### 1.1.1.1ZBO_Spd

| *Name* | *Code* |
|---|---|
| klasyfikator 1 | klasyf1 |
| klasyfikator 2 | klasyf2 |
| klasyfikator 3 | klasyf3 |
| klasyfikator 4 | klasyf4 |
| numer zlecenia | nrZlec |
| rok zlecenia | rok |
| typ zlecenia | typZlec |
| opis typu zlecenia | opisTypZlec |
| kod sposobu zapłaty | kodSpoZaro |
| opis kodu sposobu zapłaty | opisSpoZaro |
| symbol środka transportu | srTrans |
| opis symbolu środka transportu | opisSrTrans |
| list przewozowy | listPrzew |

| | |
|---|---|
| opis listu przewozowego | opisListPrzew |
| typ węgla | typWegla |
| opis typu węgla | opisTypWegla |
| grupa sortymentu | grupaSort |
| opis grupy sortymentu | opisGrupaSort |
| zesłownikowany sortyment | sort2 |
| opis sortymentu | opisSort |
| sortyment | sort |
| sposób wzbogacania | indpl |
| nazwa sortymentu | nazwaSort |
| kaloryczność | qir |
| popiół | ar |
| wilgoć | wtr |
| siarka | sr |
| klasa węgla | klasa |
| klasa1 | klasa1 |
| klasa2 | klasa2 |
| klasa3 | klasa3 |
| numer kopalni (grupa) | grpNrKop |
| symbol kopalni (grupa) | grpSymbKop |
| nazwa kopalni (grupa) | grpNazwaKop |
| numer kopalni (kopalnia) | kopNrKop |
| symbol kopalni (kopalnia) | kopSymbKop |
| nazwa kopalni (kopalnia) | kopNazwaKop |
| numer kopalni (ruch) | ruchNrKop |
| symbol kopalni (ruch) | ruchSymbKop |
| nazwa kopalni (ruch) | ruchNazwaKop |
| numer płatnika (knt) | nrPla |
| numer odbircy (knt) | nrOdb |
| dokument | dokument |
| numer faktury | nrFakt |
| rok wystawienia faktury | rokFakt |
| numer korekty do faktury | nrFaktKor |
| rok korekty faktury | rokKorFakt |
| data ekspedycji | dataEksp |
| data kwalifikacji | dataKwal |
| data wystawienie faktury | dataFakt |
| data zapłaty za fakturę | dataZapFakt |
| tonaż | tonaz |
| cena netto | cenaNetto |
| cena brutto | cenaBrutto |
| wartość netto | wartoscNetto |
| wartość VAT | wartoscVat |
| wartość brutto | wartoscBrutto |
| stopa VAT | stVat |
| termin płatności w dniach | termPlat |
| kiedy zapłacić (klucz do es17) | kiedyZap |
| opis kiedy zapłacić | opisKiedyZap |
| numer umowy (spozaro) | nrUmowy |
| rodzaj zbytu | rodzajZbytu |
| grupa cen | grupaCen |
| nazwa stacji | nazwaStacji |
| nazwa bocznicy | nazwaBocznicy |
| kto płaci przewoźne | przewoz |
| data wstawienia wpisu | dataRep |

| | |
|---|---|
| typ replikacji | typRep |
| nazwa paczki niosacej dane | nazwaPaczki |

## 10.. Wnioski

1. Podsumowując należy stwierdzić, że dzięki rozwiązaniom klasy Business Intelligence zarządzanie przedsiębiorstwem wchodzi w nowy etap. Uciążliwość uzyskiwania sprawozdań w różnym układzie była dotychczas bolączką wielu przedsiębiorstw. W dobie zwiększania efektywności pracy angażowała wykwalifikowanych pracowników do pracy polegającej na przenoszeniu danych z miejsca w miejsce. Dzisiaj istnieje możliwość powiązania dowolnych danych, również z innych systemów, w tym uzyskanych w trybie dostępu przez internet. Business Objects umożliwia dołączanie do Świata Obiektów danych z innych baz danych a także przechowywanych w postaci plików programu Excel czy nawet tekstowych plików płaskich.

2. Dotychczasowe doświadczenia w stosowaniu rozwiązań klasy Business Inteligence potwierdzają ich skuteczność i praktyczność w zaspakajaniu potrzeb wynikających z efektywnego funkcjonowania firmy.

## Literatura

1. Praca zbiorowa. *Business Intelligence,* Wydawnictwo Libridis, 2002.
2. Ladanyi H.; *SQL – Księga Eksperta,* Wydawnictwo Heliun. Gliwice, 2000.
3. Koszowski Z., Syrkiewicz J.: *Kierunki przemian w obsłudze informatycznej górnictwa węgla kamiennego.* Wiadomości Górnicze nr 10. Katowice 1994.
4. Koszowski Z., Syrkiewicz J.: *Efektywność wdrażania Zintegrowanego Systemu Wspomagającego Zarządzania Przedsiębiorstwem – System SZYK, w sektorze górnictwa węgla kamiennego,* PTI – Efektywność zastosowań systemów informatycznych – tom III. Wydawnictwo Naukowo-Techniczne, Warszawa – Szczyrk 2002.
5. Liautaud B.: *Business Intelligence Od informacji przez wiedzę do zysków,* Warszawa 2003.
6. Syrkiewicz J., Rymaszewski S.: *Zastosowanie systemów klasy Business Inteligence oraz aplikacji korporacyjnych w górnictwie węgla kamiennego,* Szkoła Ekonomiki i Zarządzania w Górnictwie, Bukowina Tatrzańska 2003r. Wydawnictwo AGH. Kraków, 2003.

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 139–144 

# On Asymptotic Behaviour of a Binary Genetic Algorithm

Witold Kosiński[1,2], Stefan Kotowski[3], and Jolanta Socała[4]

[1] Research Center, Department of Intelligent Systems, Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warszawa, Poland
[2] Institute of Environmental Mechanics and Applied Computer Science
Kazimierz Wielki University
ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland
wkos@pjwstk.edu.pl
[3] Institute of Fundamental Technological Research, IPPT PAN
ul. Świętokrzyska 21, 00-950 Warszawa, Poland
skot@ippt.gov.pl
[4] Institute of Mathematics, Silesian University
ul. Bankowa 14, 40-007 Katowice, Poland
jsocala@ux2.math.us.edu.pl

**Abstract.** The simple genetic algorithm (SGA) and its convergence analysis are main subjects of the article. A particular SGA is defined on a finite multi-set of individuals (chromosomes) together with mutation and proportional selection operators, each of which with some prescribed probability. The selection operation acts on the basis of the fitness function defined on individuals. Generation of a new population from given one is realized by iterative actions of those operators. Each iteration is written in the form of a transition operator acting on probability vectors which describe probability distributions of all populations. The transition operator is a power of a Markovian matrix. Thanks to the theory of Markov operators [6,9,10] new conditions for asymptotic stability of the transition operator are formulated.
**Keywords**: simple genetic algorithms, proportional selection, mutation, population, Markovian matrix, asymptotic stability

## 1 Introduction

In the last two decades there has been growing interest in universal optimization methods realized by genetic and evolutionary algorithms. That algorithms use only limited knowledge about problems to be solved and are constructed on the basis of some similarity to processes realized in nature. Wide applications of those methods in practical solutions of complex optimal problems cause a need to develop theoretical foundations for them. The question of their convergence properties is one of most important issues [2,3,4,5,7,8].

## 2 Preliminaries

Genetic (GA) as well evolutionary algorithms (EG) perform multi-directional search by maintaining a population of potential solutions, called individuals, and encourage information formation and exchange between these directions. A population, i.e. a set of individuals, undergoes a simulated evolution with a number of steps. In most general case the evolution is due to an iterative action—with some probability distributions—of a composition of three operators: mutation, crossover and selection ones. If a population is regarded as a point in the space $Z$ of (encoded) potential solutions then the effect of one iteration of this composition is to move that population to another point. In this way the action of GA as well as EA is a discrete (stochastic) dynamical system. In the paper we use the term *population* in two meanings; in the first it is a finite multi-set (a set with elements that can repeat) of individuals, in the second it is a frequency vector components of which are composed of fractions, i.e. the ratio of the number of copies of each element $z_k \in Z$ to the total population size. The action of that composition is a random operation on populations.

In the paper we are concerned with a particular case of the simple genetic algorithm (SGA) in which the mutation follows the fitness proportional selection and the crossover is not present. In the case of a binary genetic algorithm (BGA) the mutation can be characterized by the bitwise mutation rate $\mu$—the probability of the mutation of one bit of a chromosome. In SGA with known fitness function the fitness proportional selection can be treated as a multiplication of each component of the frequency vector by

the quotient of the fitness of the corresponding element to the average fitness of the population. This allows to write the probability distribution for the next population in the form of the product of the diagonal matrix with the population (frequency) vector. Moreover, results of the mutation can also be written as a product of another matrix with the population (probability) vector. Finally the composition of both operations is a matrix (cf. (10)), which leads to the general form of the transition operator (cf. (12)) acting on a new probability vector representing a probability distribution of appearance of all populations of the same size equal to the population size *PopSize*. The matrix appearing there turns to be Markovian and each subsequent application of SGA is the same as the subsequent composition of that matrix with itself (cf. (13)). In the paper thanks to the well-developed theory of Markov operator [1,6,9,10] new conditions for the asymptotic stability of the transition operator are formulated and some conclusions are derived.

## 3 Frequency and Population Vector

In the case of BGA the set of individuals

$$Z = \{z_0, \ldots, z_{s-1}\},$$

are *chromosomes* and they form all binary $l$-element sequences. For a better description one orders them and the set $Z$ with $s = 2^l$, becomes a list, in which its typical element (chromosome) is of the form $z_j = \{0, 0, 1, 0, \ldots, 1, 0, 0\}$.

At first by a *population* we understand any multi-set of $r$ chromosomes from $Z$, then $r$ is the population size: *PopSize*.

**Definition 1.** *By a* frequency vector of population *we understand the vector*

$$p = (p_0, \ldots, p_{s-1}), \ \textit{where} \quad p_k = \frac{a_k}{r}, \tag{1}$$

*where $a_k$ is a number of copies of the element $z_k$.*

The set of all possible populations now understood in the second meaning as frequency vectors is

$$\Lambda = \{p \in \mathbb{R}^s : \ p_k \geq 0, \ p_k = \frac{d}{r}, \ d \in \mathbb{N}, \ \sum_{k=0}^{s-1} p_k = 1 \ \}. \tag{2}$$

When GA is realized by an action of the so-called transition operator on given population, new population is generated. Since the transition between two subsequent populations is random and is realized by a probabilistic operator, then if one starts with a frequency vector, a probabilistic vector can be obtained, in which $p_i$ may not be rational any more. Hence for our analysis the closure of the set $\Lambda$, namely

$$\overline{\Lambda} = \{x \in \mathbb{R}^s : \bigwedge k, \ x_k \geq 0, \text{ and } \sum_{k=0}^{s-1} x_k = 1 \ \}, \tag{3}$$

is more suitable.

## 4 Selection Operator

Optimization problem at hand is characterized by a goal (or cost) function. If we transform it by a standard operation to a nonnegative function we will get the so-called *fitness function* $f : Z \to \mathbb{R}^+$. If we assume the first genetic operator is the *fitness proportional selection*, then the probability that the element $z_k$ from a given population $p$ will appear in the next population equals

$$\frac{f(z_k)p_k}{\overline{f}(p)}, \tag{4}$$

where $\overline{f}(p)$ is the *average population fitness* denoted by

$$\overline{f}(p) = \sum_{k=0}^{s-1} f(z_k)p_k. \tag{5}$$

Then the transition from the population $p$ into the new one, say $q$, can be given by

$$q = \frac{1}{\overline{f}(p)}\boldsymbol{S}p, \tag{6}$$

where the matrix $\boldsymbol{S}$ of the size $s$, has on its main diagonal the entries

$$S_{kk} = f(z_k). \tag{7}$$

Matrix $\boldsymbol{S}$ describes *selection operator* [5,7,8].

## 5 Mutation Operator

The second genetic operator we consider the *binary uniform mutation* with a parameter $\mu$ as the probability of changing bits 0 into 1 or vice versa. If the chromosome $z_i$ differs from $z_j$ at $c$ positions then the probability of mutation of the element $z_j$ into the element $z_i$ is

$$U_{ij} = \mu^c(1-\mu)^{l-c}. \tag{8}$$

Then we may define a matrix

$$\boldsymbol{U} = [U_{ij}] \ ,$$

with $U_{ij}$ as in (8) and $U_{ii}$—the probability of the surviving of the element (individual) $z_i$. In general one requires

1.
$$U_{ij} \geq 0;$$

2.
$$\sum_{i=0}^{s-1} U_{ij} = 1 \ , \ \text{for all } j. \tag{9}$$

## 6 Transition Operator

When we have the specific population $p$, then it means $p$ is a frequency vector and $p \in \Lambda$. If the mutation and selection (random) operators are applied to it they could lead $p$ out the set $\Lambda$. The action of the genetic algorithm at the first and at all subsequent steps is the following: if we have a given population $p$ then we sample with returning $r$-elements from the set $Z$, and the probability of sampling the elements $z_0, \ldots, z_{s-1}$ is described by the vector $G(p)$, where

$$G(p) = \frac{1}{\overline{f}(p)}\boldsymbol{U}\boldsymbol{S}p \ . \tag{10}$$

This $r$-element vector is our new population $q$.

Let us denote by $W$ the set of all possible $r$-element populations composed of elements selected from the set $Z$, where elements in the population could be repeated. This set is finite and let its cardinality be $M$. It can be shown that the number $M$ is given by some combinatoric formula, cf. [12]. Let us order all populations, then we identify the set $W$ with the list $W = \{w^1, \ldots, w^M\}$. Typical $w^k, k = 1, 2, \ldots, M$ is some population for which we used the notation $p$ in the previous section. That population will be identified with its frequency vector or probabilistic vector. This means that for given population $p = w^k = (w_0^k, \ldots, w_{s-1}^k)$ , the number $w_i^k$, for $i \in \{0, \ldots, s-1\}$ , denotes the probability of sampling from the population $w^k$ the individual $z_i$. If $p$ is a frequency vector then the number $w_i^k$ the fraction of the individual $z_i$ in the population $w^k$.

Beginning our implementation of BGA from an arbitrary population $p = w^k$ in the next stage each population $w^1, \ldots, w^M$ can appear with the probability, which can be determined from our analysis. In particular, if in the next stage the population has to be $q$, with the position $l$ on our list $W$ (it means $q = w^l$), then this probability [7,11,12] is equal

$$r! \prod_{j=0}^{s-1} \frac{\left(\mathcal{G}(p)_j\right)^{rq_j}}{(rq_j)!} \ . \tag{11}$$

After two steps, every population $w^1, \ldots, w^M$ will appear with some probability, which is a double composition of this formula. It will be analogously in the third step and so on. This formula gives a possibility of determining all elements of a matrix $\boldsymbol{T}$ which defines the probability distribution of appearance of populations in the next steps, if we have current probability distribution of the populations. With our choice of denotations for the populations $p$ and $q$, the element $(l, k)$ of the matrix will give transition probability from the population with the number $k$ into the population with the number $l$. It is important that elements of the matrix are determined once forever, independently of the number of steps. The transition between elements of different pairs of populations is described by different probabilities (11) represented by different elements of the matrix. We can see that the nonnegative, square matrix $\boldsymbol{T}$ of dimension $M$, with elements $p_{lk},\ l, k = 1, 2, \ldots, M$ has the property: the probability distribution of all $M$ populations in the step $t$ is given by the formula

$$\boldsymbol{T}^t u \quad t = 0, 1, 2, \ldots$$

Let us denote by

$$\Gamma = \{x \in \mathbb{R}^M : \forall k\ x_k \geq 0 \text{ oraz } ||x|| = 1\},$$

where $\|x\| = x_1 + \ldots + x_M$, for $x = (x_1, \ldots, x_M)$, the set of new $M$-dimensional probabilistic vectors. A particular component of the vector $x$ represents the probability of the appearance of this population from the list $W$ of all $M$ populations. The set $\Gamma$ is composed of the all possible probability distributions for $M$ populations. Then described implementation transforms, at every step, the set $\Gamma$ into the same.

Notice that if at the beginning we start our SGA at a specific population $p$, which attains the place $j$-th on our list $W$,i.e. $p = w^j$ , then the vector $u$ will denote the particular situation of the population distribution in the step zero 0 if

$$u = (0, \ldots, 0, 1, 0 \ldots, 0) \in \mathbb{R}^m.$$

On the set $\Gamma$ the basic, fundamental *transition operator*,

$$T(\cdot) : \mathbb{N} \times \Gamma \to \Gamma. \tag{12}$$

is defined. According to the above remark the transition operator $T(t)$ is linked with the above matrix by the dependence

$$T(t) = \boldsymbol{T}^t\ . \tag{13}$$

If $u \in \Gamma$, then $T(t)u = \big((T(t)u)_1, \ldots, (T(t)u)_M\big)$ is the probability distribution for $M$ populations in the step number $t$ , if we have begun our implementation of SGA given by $\mathcal{G}$ (10) from the probability distribution $u = (u_1, \ldots, u_M) \in \Gamma$, by $t$ - application of this method. The number $(T(t)u)_k$ for $k \in \{1 \ldots, M\}$ denotes the probability of appearance of the population $w^k$ in the step of number $t$. By the definition $\mathcal{G}(p)$ in (10),(11), and the remarks made at the end of the previous section, the transition operator $T(t)$ is linear for all natural $t$.

Notice that though the formula (11) determining individual entries (components) of the matrix $\boldsymbol{T}$ is a population dependent, and hence nonlinear, the transition operator $T(t)$ is linear thanks to the order relation introduced in the set $W$ of all $M$ populations. The multi-index $l, k$ of the component $p_{lk}$ kills, in some sense, this nonlinearity, since it tells (is responsible)for a pair of populations between which the transition takes place. The matrix $\boldsymbol{T}$ is a Markovian matrix. This fact permits us to apply Theory of Markov operators to analyze the convergence of genetic algorithms [1,6,9,10].

Notice that the action of the matrix $\boldsymbol{T}$ can be seen as follows. In the space of the all possible populations there is a walking point, which attains its next random position numbered by $1, 2, \ldots, M$, as an action of SGA on the actual population, with probabilities $u_1, u_2, \ldots, u_M$. We know that if at the moment $t$ (in the generation number $t$) we had population $p$ with the position $k$ on our list, i.e. the population $w^k$, then the probability that at the moment $t + 1$ (in the generation number $t + 1$) it will attain population $q$ with the position $l$, on our list, i.e. the population $w^l$, is $p_{lk}$, and this probability is independent of the number of steps in which it is realized. With this denotation the probability $p_{lk}$ is given by the formula (11).

Let $e_k \in \Gamma$ be a vector which at the $k$-th position has one and zeroes at the other positions. Then $e_k$ describes the probability distribution in which the population $w^k$ is attained with the probability 1.

By the notation $T(t)w^k$ we will understand

$$T(t)w^k = T(t)e_k \tag{14}$$

which means that we begin the GA at the specific population $w^k$ .

Further on we will assume $U_{jj} > 0$ for $j \in \{0, \ldots, s-1\}$. Notice that in the case of binary mutation (8) this condition will be satisfied if $0 \leq \mu < 1$.

**Definition 2.** *We will say that the model is* asymptotically stable *if there exists* $u^* \in \Gamma$ *such that:*

$$T(t)u^* = u^* \quad \text{for} \quad t = 0, 1, \ldots \tag{15}$$

$$\lim_{t\to\infty} ||T(t)u - u^*|| = 0 \quad \text{for all } u \in \Gamma \ . \tag{16}$$

Since for $k \in \{1, \ldots, M\}$ we have

$$|\big(T(t)u\big)_k - u_k^*| \leq ||T(t)u - u^*|| \ , \tag{17}$$

then (16) will gives

$$\lim_{t\to\infty} \big(T(t)u\big)_k = u_k^* \ . \tag{18}$$

It means that probability of appearance of the population $w^k$ in the step number $t$ converges to a certain fixed number $u_k^*$ independently of the initial distribution $u$ . It is realized in some special case, when our implementation begun at one specific population $p = w^j$ .

We shall say that from the chromosome $z_a$ it is possible to obtain $z_b$ in one mutation step with a positive probability if $U_{ba} > 0$. We shall say that from the chromosome $z_a$ it is possible to get the chromosome $z_b$ with positive probability in $n$-step mutation if there exists a sequence of chromosomes $z_{i_o}, \ldots, z_{i_n}$, such that $z_{i_o} = z_a$ , $z_{i_n} = z_b$ and any $z_{i_j}$ for $j = 1, \ldots, n$ is possible to attain from $z_{i_{j-1}}$ in one step with a positive probability.

**Definition 3.** *Model is* pointwise asymptotically stable *if there exists such a population* $w^j$ *that*

$$\lim_{t\to\infty} (T(t)u)_j = 1 \text{ for } u \in \Gamma \ . \tag{19}$$

Condition (19) denotes that in successive steps the probability of appearance of another population than $w^j$ tends to zero. It is a special case of the asymptotic stability for which

$$u^* = e_j.$$

**Theorem 1.** *Model is pointwise asymptotically stable if and only if there exists exactly one chromosome* $z_a$ *with such a property that it is possible to attain it from any chromosome in a finite number of steps with a positive probability. In this situation the population* $w^j$ *is exclusively composed of the chromosomes* $z_a$ *and*

$$T(t)w^j = w^j \tag{20}$$

*holds. Moreover, the probability of appearance of other population than* $w^j$ *tends to zero in the step number* $t$ *with a geometrical rate, i.e. there exists* $\lambda \in (0, 1)$, $D \in \mathbb{R}_+$ *thats*

$$\sum_{\substack{i=1 \\ i\neq j}}^{M} (T(t)u)_i \leq D \cdot \lambda^t \ . \triangle \tag{21}$$

The proofs of our theorems and auxiliary lemmas are stated in original articles [12,13].

Numbers $\lambda$ and $D$ could be determined for a specific model. It will be the subject of the next articles.

Theorem 1 states that the convergence to one population could occur only under specific assumptions. This justifies the investigation of asymptotic stability that in Definition 2.

**Definition 4.** *By an* attainable chromosome *we denote* $z_a \in Z$ *such that it is possible to attain it from any other chromosome in a finite number of steps with a positive probability. Let us denote by* $Z^*$ *the set of all* $z_a$ *with this property.*

**Theorem 2.** *Model is asymptotically stable if and only if* $Z^* \neq \emptyset$. $\triangle$

**Theorem 3.** *Let us assume that the model is asymptotically stable. Then the next relationship holds:*

*(war)* $u_k^* > 0$ *if and only if the population* $w^k$ *is exclusively composed of chromosomes belonging to the set* $Z^*$. $\triangle$

## 7 Conclusions

Here we set the summary of our results obtained here and in our other papers [11,12,13]:

1. If $Z^* = \emptyset$ then there is a lack of asymptotic stability.
2. If $Z^* \neq \emptyset$ then asymptotic stability holds but:
3. If cardinality $(Z^*) = 1$ then pointwise asymptotic stability (in some sense convergence to one population) holds.
4. If cardinality $(Z^*) > 1$ then asymptotic stability holds, but there is no pointwise asymptotic stability.
5. If $Z^* = Z$ then $u_k^* > 0$ for all $k \in \{1, \ldots, M\}$.

Remark. In SGA with a positive mutation probability, it is possible to attain any individual (chromosome) from any other individual. Then there is more than one chromosome which is possible to attain from any other in a finite number of steps with a positive probability. Hence, by Theorem 1, it is impossible to get the population composed exclusively of one type of chromosome.
The last conclusion means that if any chromosome is possible to attain from any other in a finite number of steps with a positive probability then in the limit (probability distribution) of infinite number of generations each population (has a positive probability) may be reached with a positive probability.
**Acknowledgement** The research work on the paper was partially done by W.K and S.K. in the framework of the KBN Project (State Committee for Scientific Research) No. 3 T11 C007 28. Authors thanks Professor Zbigniew Michalewicz for the inspiration and discussions.

## References

1. A. Lasota, *Asymptotyczne własności półgrup operatorów Markowa,* Matematyka Stosowana. Matematyka dla Społeczeństwa, **3** (45), 2002, 39–51.
2. P. Kieś i Z. Michalewicz, *Podstawy algorytmów genetycznych,* Matematyka Stosowana. Matematyka dla Społeczeństwa, **1** (44), 2000, 68–91.
3. Z. Michalewicz, *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa, 1996.
4. J. Cytowski, *Algorytmy genetyczne: podstawy i zastosowania*, Seria: Problemy Współczesnej Nauki —Teoria i Zastosowania, Nr **18,** Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1996.
5. M. D. Vose, *The Simple Genetic Algorithm: Foundation and Theory*, MIT Press, Cambridge, MA, 1999.
6. A. Lasota, J. A. Yorke, *Exact dynamical systems and the Frobenius-Perron operator,* Trans. Amer. Math. Soc. **273** (1982), 375–384.
7. J. E. Rowe, *The dynamical system models of the simple genetic algorithm,* in Theoretical Aspects of Evolutionary Computing, Leila Kallel, Bart Naudts, Alex Rogers (Eds.), Springer, 2001, pp.31–57.
8. R. Schaefer, *Podstawy genetycznej optymalizacji globalnej*, Wydawnictwo Uniwersytetu Jagielońskiego, Kraków 2002.
9. R. Rudnicki, *On asymptotic stability and sweeping for Markov operators,* Bull. Polish Acad. Sci. Math., **43** (1995), 245–262.
10. J. Socała, *Asymptotic behaviour of the iterates of nonnegative operators on a Banach lattice,* Ann. Polon. Math., **68** (1), (1998), 1–16.
11. J. Socała, W. Kosiński, S. Kotowski, *O asymptotycznym zachowaniu prostego algorytmu genetycznego,* Matematyka Stosowana. Matematyka dla Społeczeństwa, **6** (47), 2005, 70–86.
12. J. Socała, *Markovian approach to genetic algorithms,* under preparation.
13. S. Kotowski, J. Socała, W. Kosiński, Z. Michalewicz, *Markovian model of simple genetic algorithms and its asymptotic behaviour,* sumitted for publication, 2005.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp. 145-151
ISBN 83-922646-0-6


# Comparative Analysis of Reporting Mechanisms Based on XML Technology

Dariusz Król[1], Jacek Oleksy[2], Małgorzata Podyma[2] and Bogdan Trawiński[1]

[1] Wrocław University of Technology, Institute of Applied Informatics,
Wybrzeże S. Wyspiańskiego 27, 50-370 Wrocław, Poland
[2] Computer Association of Information BOGART Ltd.,
Rejtana 9-11, 50-015 Wrocław, Poland
[1] {dariusz.krol, trawinski}@pwr.wroc.pl, [2] {oleksy, podyma}@bogart.wroc.biz

**Abstract.** Comparative analysis of reporting mechanisms based on XML technology is presented in the paper. The analysis was carried out as the part of the process of selecting and implementing of reporting mechanisms for a cadastre information system. The reports were designed for two versions of the system, i.e. for the internet system based on PHP technology and the fat client system in two-layer client-server architecture. The reports for the internet system were prepared using XSLT for HTML output and using XML-FO for PDF output and compared with reports implemented using Free PDF library. Each solution was tested by means of the Web Application Stress Tool in order to determine what limits in scalability and efficiency could be observed. As far as the desktop system is concerned three versions of reporting mechanisms based on Crystal Reports, Microsoft Reporting Services and XML technology were accomplished and compared with the mean execution time as the main criterion.

## 1. Introduction

On the market there are many commercial reporting tools. They can take data from various sources such as ODBC, JDBC, EJB, OLE DB, Oracle, DB2, Microsoft SQL Server, Access, BDE, XML, txt and others. Almost all of them allow to export reports to such file formats as html, pdf, xml, rtf, xls, csv, txt and some to doc, LaTeX, tiff, jpeg, gif [8], [10]. The study presented in the paper was carried out in order to choose the most suitable reporting mechanisms for a new version of a cadastre system used by many local governments in Poland. Selection of a reporting tool for an information system is a difficult task. It should be accomplished carefully and many factors should be taken into account such as user expectations, ability of report designing and modifying, possibility of presenting complex data in a readable way, costs and licensing as well as the performance and scalability required. However just the prices usually have decisive influence on the selection of reporting tools for the cadastre system. Commercial tools are perceived by local governments to be too expensive and this was the main reason for that we started to design and programme reporting mechanisms based on low level open source tools.

The selection process should also take into account results of feature comparative analyses [5, 10] and benchmark tests carried out on reporting tools by their producers or by independent organizations [1, 2, 3, 4]. All benchmark tests have proved that such commercial tools as Actuate iServer, Crystal Enterprise and Cognos ReportNet significantly exceed the requirements of the cadastre system considered in the paper. Those benchmark tests were accomplished using highly efficient and expensive hardware and were designed for the developers who want to build large-scale applications. Therefore the results were not usable for the decision process of selecting the most convenient reporting tool for the cadastre system. This also led us to start testing the performance and scalability of our solution.

The most prospective seem to be mechanisms based on XML language. Using the DOM (Document Object Model) you can create and manipulate XML documents by your application with dynamic access to the content of XML documents. XSL and XSL-FO languages enable you to create style sheet documents which control the presentation of a XML document and describe the transformation of a XML document into other formats as HTML, SVG, CSV, plain text or PDF [6, 7, 9].

The model of generating reports using XML language is shown in Fig. 1. XSLT processor takes XML and XSLT documents and transforms them into new XML document or into a document in another format for

example into HTML document. In order to obtain PDF output XSL-FO document is created and then transformed by FOP processor into final PDF document.

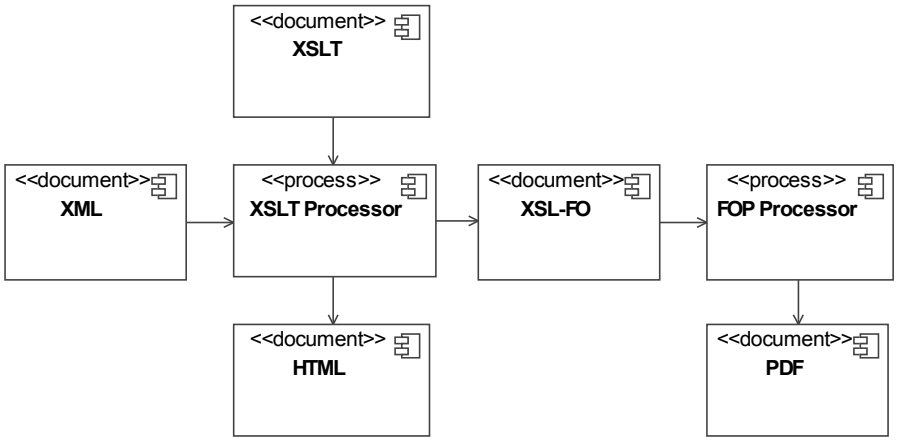



**Fig 1. Model of creating reports using XML language**

## 2. Reports in the Cadastre System

The maintenance of real estate cadastre registers is dispersed in Poland. There are above 400 information centres located by district local self-governments as well as by the municipalities of bigger towns which exploit different cadastre systems. The majority of cadastre systems in Poland are developed in two-layer client-server architecture and deployed on Microsoft SQL Server or Oracle database management systems running on Intel servers. The EGB2000 system for which we were looking for the best reporting tools is deployed in above 100 local governments throughout Poland while the EGB2000-INT system providing an internet access to cadastral databases is used in about 50 intranets and extranets. The EGB2000 system was implemented in fat client architecture using Visual C++ in Visual Studio.NET environment whereas the EGB2000-INT system was programmed using PHP script language and accommodated for the cooperation with Apache or IIS web servers.

Four years of developing and maintaining cadastre systems in many information centres in Poland experienced us that users regard reporting functions as the most important in the systems and require reporting tools of high quality. They expect reports presenting complex data clearly, allowing to select data using various criteria, enabling to change the scope of data shown in a flexible way and taking time as shortest as possible to generate.

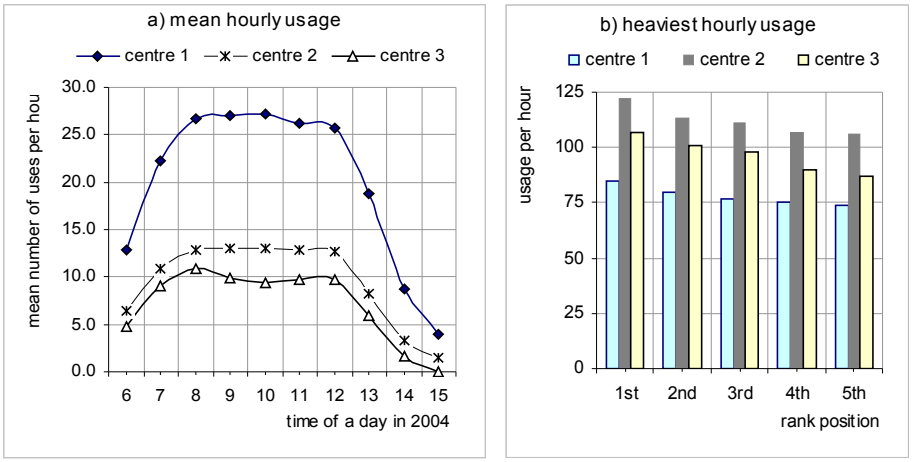



**Fig. 2. Hourly usage of reports in internet system in selected centres**

There are 117 reports available in the fat client system and 15 in the internet one. In order to investigate what and how the reports are used in both systems report usage logs are maintained. The analysis of these logs in six selected centres indicated that three groups of reports were the most frequently employed by the users of the systems. These are extracts from registers, notifications of changes and various lists of objects with their attributes. Fig. 2a presents mean hourly usage of reports in the internet system in 2004. The five best results from the top for each centre studied are shown in Fig. 2b. Data indicate that the maximum hourly usage reached 122 what equals to only 2 reports per one minute. Mean hourly usage of the reports in fat client system during June 2005 for three another centres is shown in Fig. 3a. Analogously to Fig. 2b the top five results are placed in Fig. 3b. Data show that the maximum hourly usage reached 150 what equals to only 2.5 per minute. The load of both systems was rather moderate.

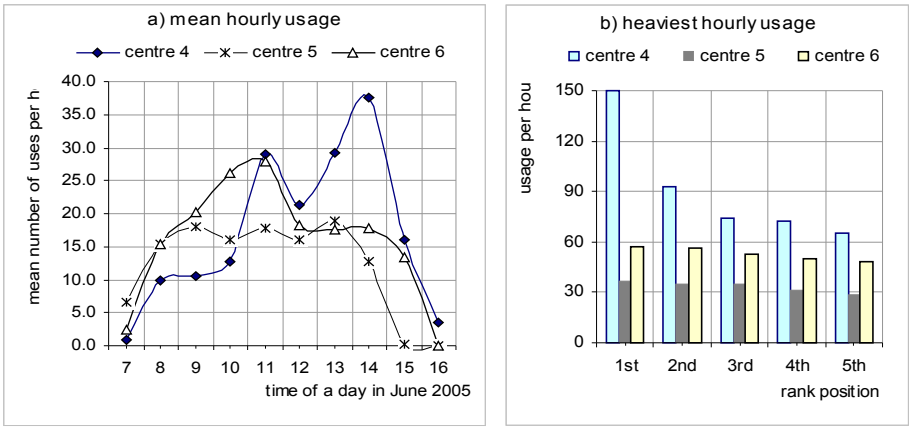



**Fig. 3. Hourly usage of reports in fat client system in selected centres**

## 3. Testing of Reporting Mechanisms Implemented in the internet Cadastre System

Three versions of reporting mechanisms implemented in the internet system were compared, i. e. one based on XML technology with HTML output (HTML), the second employing FOP processor to generate reports in PDF format (FOP) and the third programmed using Free PDF library to produce PDF output (FPDF). Each solution was tested using Web Application Stress Tool in order to determine what limits in scalability and efficiency could be observed. The most frequently used report i.e. an extract from land register unit was the main subject of all tests.

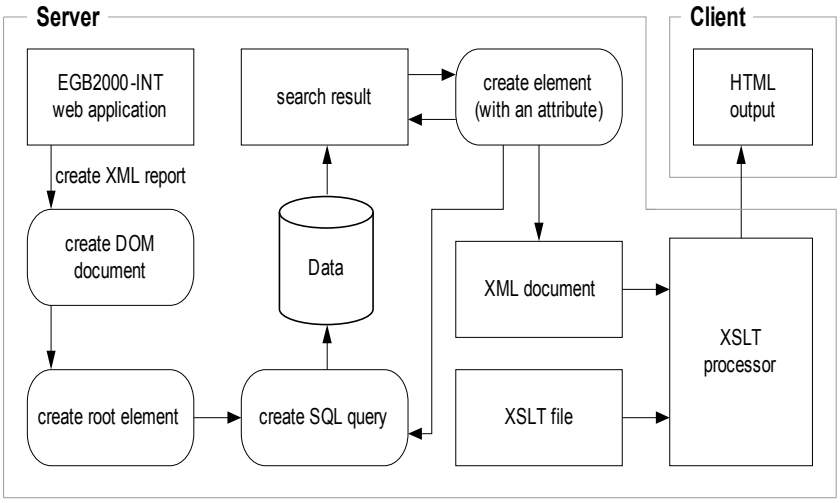



**Fig. 4. Architecture of XML reporting mechanism for internet system with HTML output**

Architectures of the first two reporting mechanisms are presented in Fig 4 and Fig. 5. In both cases the system applications create XML documents using DOM and selecting records from database according to the search criteria input by users. Then required output is generated by XSLT processor or by FOP.

In Fig. 6 the architecture of the third reporting mechanism is shown. The Free PDF library generates a PDF file directly on the basis of the result of SQL query and then saves it on the server disk and sends it to the client's browser. When the client finishes working with the report the PDF file is deleted from the server disk.

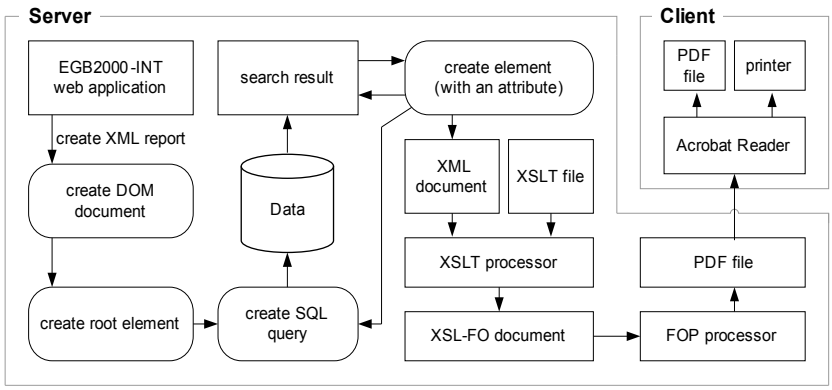



**Fig. 5. Architecture of XML reporting mechanism for internet system using FOP**

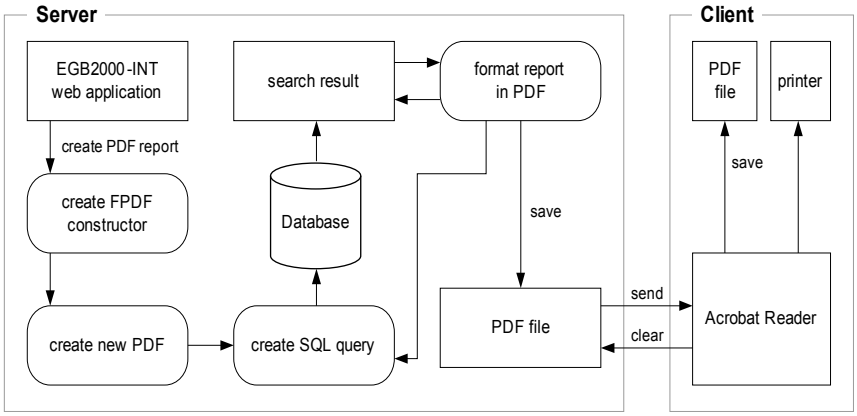



**Fig. 6. Architecture of reporting mechanism using Free PDF Library**

### 3.1. Testing of mechanisms based on XML providing HTML and PDF output

The experiment was carried out using the server with 2.0 GHz Intel Pentium IV and 1 GB RAM, XP Windows Professional operating system, Apache web server and MS SQL Server 2000 as DBMS. The tests simulated continuous work of 1, 2, 5 or 10 concurrent users who generated reports containing 1, 10, 100 or 500 parcels. The results are presented in Fig. 7 where -10 and -100 denote the numbers of parcels comprised in respective reports. It can be easily seen that the mechanism using only XSLT processor gives better results than the mechanism having additional FOP processor. In all tests for one user mean response time was shorter than 1 s and for 10 users smaller than 1.6 s. However 15 users or reports with 500 parcels caused the load too big for the operating system. It started to suspend or to display out of memory error.

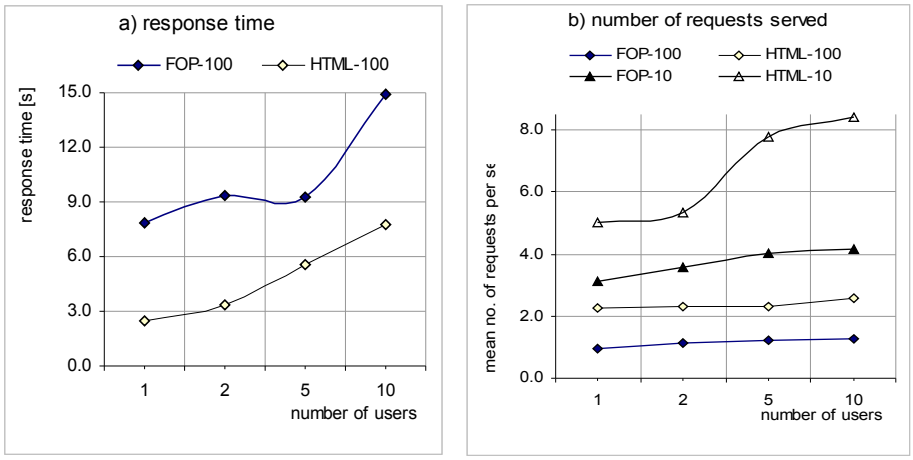



**Fig. 7. Test results of XML reporting mechanism with HTML output and using FOP**

### 3.2. Testing of mechanisms using FOP and Free PDF Library

In the experiment 2.0 GHz Intel Pentium IV with 1 GB RAM was used as the server. Windows 2000 Professional operating system (different from OS used in sec. 3.1), Apache web server and MS SQL Server 2000 as DBMS were installed.

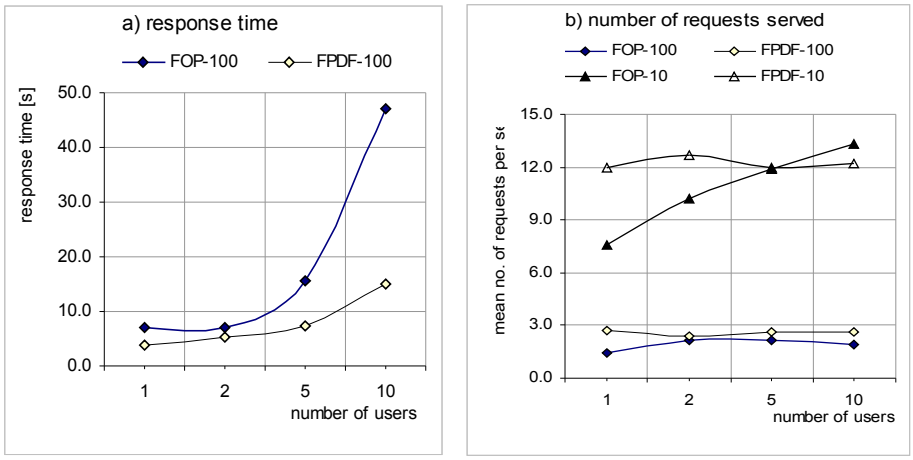



**Fig. 8. Test results of reporting mechanisms based on FOP and Free PDF library**

Analogously the tests simulated continuous work of 1, 2, 5 or 10 concurrent users who generated reports containing 1, 10, 100 or 500 parcels. Denotation of mechanisms in Fig. 8 is similar to that used in Fig. 7. The results in Fig. 8 revealed better performance of the mechanism using Free PDF library, but the difference was not big enough to compensate longer time and inconvenience of programming. The tests for 15 users or reports with 500 parcels also caused so heavy load that the operating system did not manage to serve it.

## 4. Testing of Reporting Mechanisms Implemented in the fat client Cadastre System

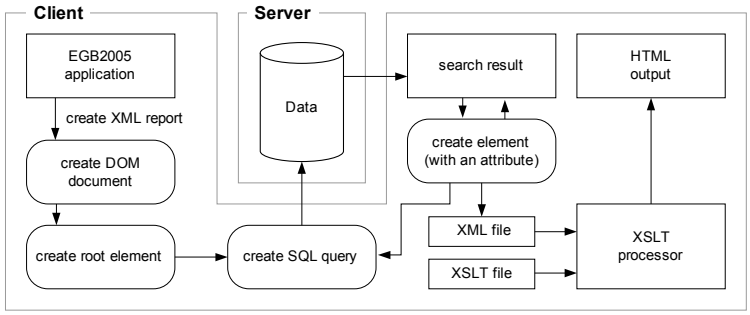



**Fig. 9. Architecture of reporting mechanism in fat client cadastre system**

In order to study the XML technology (XML) in the fat client cadastre system two additional versions of reporting mechanisms were implemented, i.e. one using Crystal Reports 10 (CR10) and the second based on Microsoft Reporting Services (MRS). In Fig. 9 the architecture of reporting mechanism using XML technology implemented in the fat client cadastre system is presented.

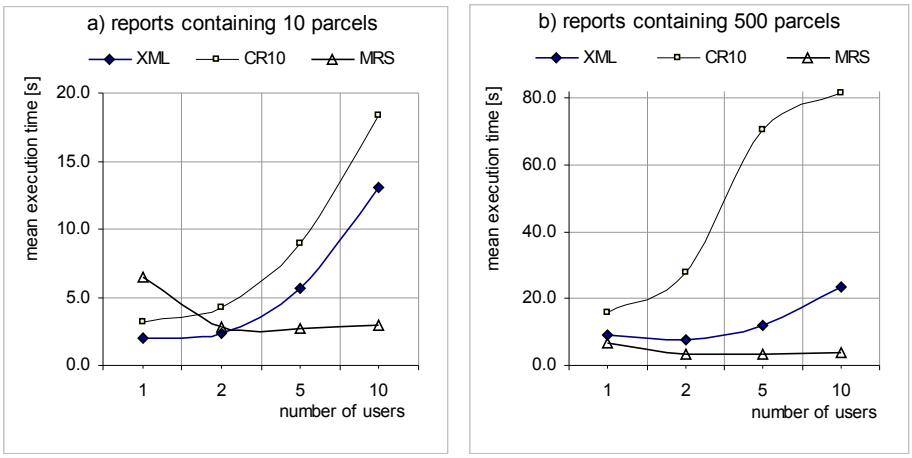



**Fig. 10. Test results of reporting mechanisms in fat client cadastre system**

Each solution was tested with the most frequently used report i.e. an extract from land register unit. The tests simulated continuous work of 1, 2, 5 or 10 concurrent users who generated 20 the same reports containing 1, 10, 100 or 500 parcels. Mean time of generating individual reports was calculated. The MRS mechanism gave the best results (see Fig. 10) and the XML mechanism showed better performance than the CR10 one. Perhaps it was caused by the method of designing CR10 reports, which contained SQL expressions in rpt files.

## 5. Conclusions

All the tests were carried out using the configuration of hardware and software common for majority of information centres. They proved that all reporting mechanisms tested could fulfil present system requirements of performance. Although the mechanism based on Microsoft Reporting Services revealed the best performance, it should be treated cautiously. It is relatively new and not all features and behaviour are known yet. Moreover we do not suppose that our clients having Oracle databases would maintain Microsoft SQL Server as their second database environment. In turn Crystal Reports is limited to one level of sub-reports what considerably makes it more difficult to design reports selecting complex data. The most prospective seems to be XML technology which has no drawbacks of other mechanisms tested taking into account all present aspects of

exploiting the cadastre information system in local governments. We appreciated wide scope of XML language functions and especially the ability to separate content processing form presentation layer and the dynamic access to the content of a XML document. The XML technology turned out to be especially useful for the internet version of the cadastre system as far as both technical and economical factors are concerned. In fact the tests revealed the limits of XML based reporting tools, but those limits are at present far beyond the real usage of the system.

## References

1. *Actuate 7 iServer vs. Crystal Enterprise 9. An Interactive Reporting Performance Benchmark Project Final Report,* National Technical Systems (2003).
2. *Actuate 7SP1 Performance and Scalability on Windows 2003,* Actuate Corporation (2004).
3. *Cognos ReportnNet Scalability Benchmarks – Microsoft Windows,* Cognos (2004).
4. *Crystal Enterprise 9.0 Baseline Benchmark: Windows Server 2003 on IBM@server xSeries 16-way,* Crystal Decisions (2003).
5. *Feature comparison. Microsoft SQL Server 2000 Reporting Services vs. Business Objects Crystal Reports,* Crystal Enterprise. Certia Business Intelligence Team (2004).
6. Holzer S.: *Inside XML,* New Riders Press (2000).
7. Holzer S.: *Inside XSLT.* Que (2001).
8. Król D., Podyma M., Trawiński B.: *Selection and testing of reporting tools for an Internet Cadastre Information System,* Software Engineering: Evolution and Emerging Technologies Ed. K. Zielinski and T. Szmuc. IOS Press (2005).
9. Pawson D.: *XSL-FO,* O'Reilly (2002).
10. Podyma M.: *Review and comparative analysis of reporting tools for internet information systems,* M. Sc. Thesis (in Polish). Wrocław University of Technology (2005).

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 153–157 

# Visualization as a method for relationship discovery in data

Halina Kwasnicka[1], Urszula Markowska-Kaczmar[1], and Jacek Tomasiak

Department of Computer Science, Wroclaw University of Technology, Poland
[halina.kwasnicka; urszula.kaczmar]@pwr.wroc.pl

**Abstract.** Visualization techniques are especially relevant to multidimensional data, the analysis of which is limited by human perception abilities. The paper presents a hybrid method of multidimensional data analysis. The main goal was to test the efficiency of the method in the context of real-life medical data. A short survey of issues and techniques concerned with data visualization are also included.

## 1 Introduction

Analysis of real data sets is a very difficult task for the humans. It is extremely hard when examined data are represented by multidimensional vectors (records). The human mind is limited to the perception in two or three dimensional space. Therefore visualisation techniques lie in the area of interest of the data mining domain. A visualization process can be divided into several stages (Fig. 1). Rough data have to be preprocessed to the convenient form. The data presented in the form of multidimensional vectors are scaled or normalized. The next phase is a projection from multidimensional space into two dimensional (2D) space [5, 1]. We can distinguish linear (Principal Component Analysis, Principal Coordinate Analysis, Discriminant Coordinates) and nonlinear methods (Self Organizing Map – SOM, Unified Distance Matrix, autoassociative neural networks, Non-Linear Mapping – for example well known Sammon mapping [9], triangulation [6], SAMANN [4]). Coordinates of points obtained after projection are subsequently plotted giving the final image. Different techniques used in this stage are presented in [3] and [7]. The source data can be used to bring in the final image a bit of information which was lost during the projection. In Fig. 1 this process is shown as a distinction.

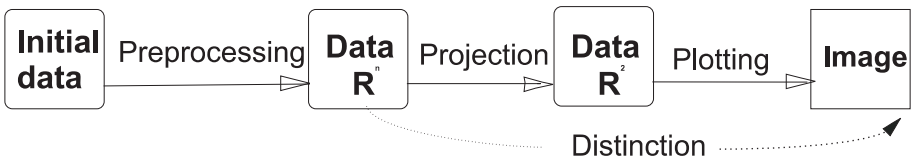



**Fig. 1.** Stages of a visualization process

Our hybrid method called *MedViz* combines known visualisation techniques. Using a neural network for Sammon mapping is a novelty of the method.

## 2 The *Medviz* method

According to the general scheme (Fig. 1), the data should be preprocessed before visualization. First, they have to be presented as a vector of numbers, with the relationship of partial order for each component. These vectors can be treated as points in the multidimensional Cartesian space ($\Re^n$).

The next step is normalization. Its aim is to obtain the distribution of values with mean $\approx$ 0,0 and variance $\approx$ 1,0 for each attribute (similar to $N(0;1)$). Thanks to this process, attributes with large (e.g. thousand) and small (e. g. one) values are treated similarly. *Medviz* contains a number of popular normalization methods, the first one is *statistical normalization* in which each value is processed according to the following equation

$$x_{\text{norm}} = \frac{x - \overline{x}}{s} \tag{1}$$

where: $x$ the original value, $\overline{x}$ the average value of the attribute, $s$ the standard deviation of the given attribute. The next one is *simple scaling* to $\langle -1.0; 1.0 \rangle$:

$$x_{\text{norm}} = \frac{x - \frac{1}{2}(x_{\max} + x_{\min})}{\frac{1}{2}(x_{\max} - x_{\min})} = \frac{2x - x_{\max} - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

where: $x$ is the original value, $x_{\min}$—the minimal value of a given attribute and $x_{\max}$—the maximal value of the attribute.

The next stage is the data projection by Sammon mapping. It is nonlinear projection which retains the distance between pairs of training patterns. The general idea lies in a minimisation of the error function, defined as follows:

$$E = \frac{1}{\sum_{j=1}^{N} \sum_{i=1}^{j} D_{ij}} \sum_{j=1}^{N} \sum_{i=1}^{j} \frac{(D_{ij} - d_{ij})^2}{D_{ij}} \tag{3}$$

where: $D_{ij}$ the distance between $i$-th i $j$-th points in the source data, $d_{ij}$ the distance between $i$-th i $j$-th points in the output data set, $N$ the number of patterns in the input data set.

The above function describes the sum of the differences of distances calculated for each pair of points. Each distance is divided by $D_{ij}$ what ensures insensibility to the scaling method. For a given set of multidimensional data the output set of points in 2D space is created in this way that $i$-th point in the output data (2D space) is an image of $i$-th point in the input data (multidimensional space). Mostly the classic euclidian distance is used. In successive iteration, points in the space move in this way that the value of the error is minimised. Usually in this process a simple Newton gradient method is applied [9].

After the projection, the image of the data is created by simplified method *Glyph Plot*. Each element of the data set is represented as one *object* (point). The object has a form of a circle with a variable diameter and colour. These parameters represent the value of the same component (attribute). The component used to a distinction can be chosen interactively. The *MedViz* method has two possibilities of the point colouring: in a temperature scale of colour (blue–green–yellow–orange–red) or in a gray scale.

In *MedViz*, for the point in the image indicated by a mouse, it is possible to obtain a bit of information, like a label and an average value (of the actual selected attribute). The selection of point and click by the mouse gives information about the neighbourhood. Double click switches the user on the table with data. In this case it is possible to get a quick look in the values of attributes of the selected record. To facilitate an analysis of a huge data set it is also possible to zoom the selected fragment of image. This function helps in the case of some small data sets. Nonuniform distribution on the space causes that the analysis can be difficult. An enlargement of the part containing interesting subset of points makes the observation much easier.

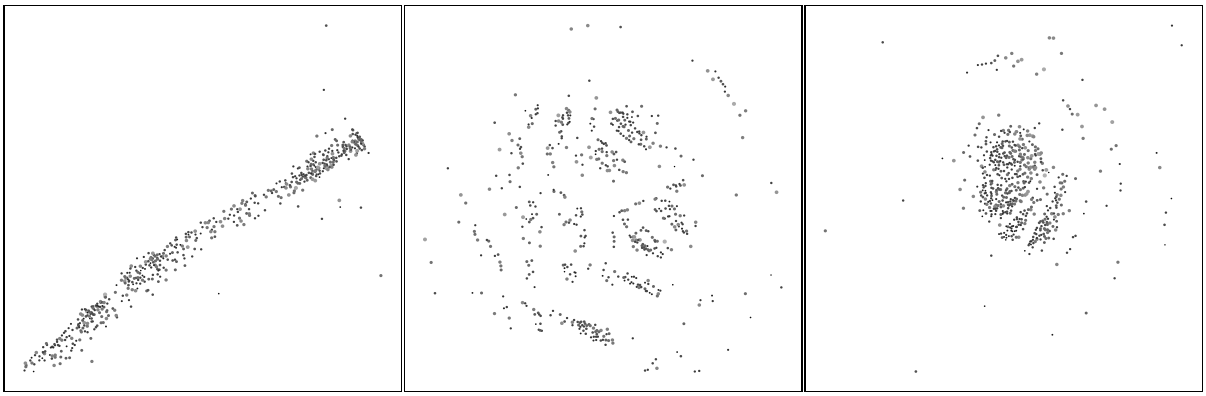
Because of the complexity of class $O(N^2)$ of Sammon mapping, the time of calculation for huge data sets becomes unacceptable. The similar situation arrives when a new record (vector) has to be visualised. Some solutions of this problem are described in ([8]).

In our method we use a simple back propagation (BP) neural network (NN) with momentum. In the output layer of the NN a linear activation function is used because new values can lie outside the range $\langle -1,0; 1,0 \rangle$. At the beginning, a part of original multidimensional vectors is projected to two dimensional space according to the Sammon mapping method. They constitute a training set for the NN realizing Sammon mapping. The error of the mapping is calculated as the sum of distances of the output vectors from the training patterns. After training the rest of original multidimensional vectors, as well as new vectors when necessary, are projected by the NN. The new (added) points are visualised as empty circles. The unknown components are automatically supplied according to the method described in the previous section. They are distinguished in the image by the colour besides the assumed scale.

## 3 Experimental studies

The aim of the experimental studies was the evaluation of an efficiency of the implemented method in data analysis in order to find relationships between attributes. Additionally, the influence of different methods of normalization was investigated. If there is no other explanation, the following values of parameter are set: number of iteration of Sammon mapping – 500, simple scaling as a normalization method, parameters of NN: number of layers – 2, min error – 0,4%, momentum – 0,9, learning coefficient – 0,001, max number of iteration – 3000.

We have experimented with three data sets: *y14c*—synthetic data set [2], *benthic* [7]—it contains information about the water purity in Chesapeake Bay and *cervix*—it has information about patients with *carcinoma of the cervix uteri*. In Fig. 2 we show the results obtained for *cervix* appropriately (a) without normalization, (b) with simple scaling, (c) with statistical normalization. The *cervix* data set in fig. 2 is a set of real data, which were not specially prepared before.



(a) (b) (c)

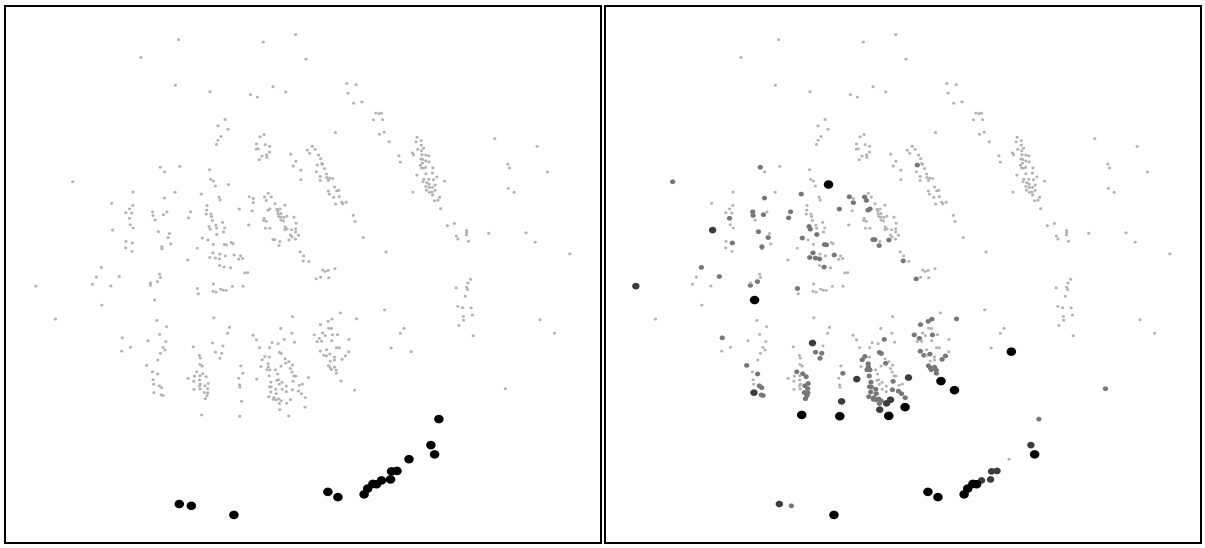
**Fig. 2.** The comparison of the normalization methods (data set *cervix*)

It is easy to notice that there is big difference in the plots obtained with the different normalization methods. The values of attributes differ a lot what can be observed in fig. 2.a for the data which were not normalized. The oblong shape of points means that values of one attribute significantly differed from other values have dominated projection process. The plot obtained by simple scaling the set of points (fig. 2.b) has more groups than after statistical normalization (fig. 2.c). At the same time statistical normalization demonstrates better capability to show outliers. In general, we can say that the only way to choose the method of normalization is to experiment with them.

The successive experiments tested the relationship between geometric distribution of points in the two dimensional space and the values of attributes. The criterion of colouring was joined with the values of respective attributes. We used two data sets: *benthic* and *cervix*. The obtained plots were analysed with respective attributes after the appropriate colouring of points (*the Glyph Plot method*). It was the basis for searching similarity between the distribution of points and distribution of colours in the different plots. The first two images (fig. 3 and 4) present data set *cervix* showed by the perspective of the attribute $P8.2$—the application of radiotherapy (External Pelvic RT), $P10$—general evaluation of the therapy efficiency, $P7.3$—an application of surgery operation (Radical Abdominal Hysterectomy without Pelvic/Paraortic Lymphadenectomy) and $P11$—the time between diagnosis and the desertion of a hospital. In Fig. 3.a we see the distinct group of points. It is easy to observe similar tendency with the other attributes. For example in Fig. 3.b the resemble group of points can be found. It represents patients with a good efficiency of the radiotherapy (bottom part of image). In this way we can conclude that radiotherapy (External Pelvic RT) has a positive influence on the efficiency of treatment. In Fig. 4, the similar relationship can be noticed between attributes $P7.3$ and $P11$. This means that patients after surgery operation Radical Abdominal Hysterectomy without Pelvic/Paraortic Lymphadenectomy, need the longer stay in a hospital.

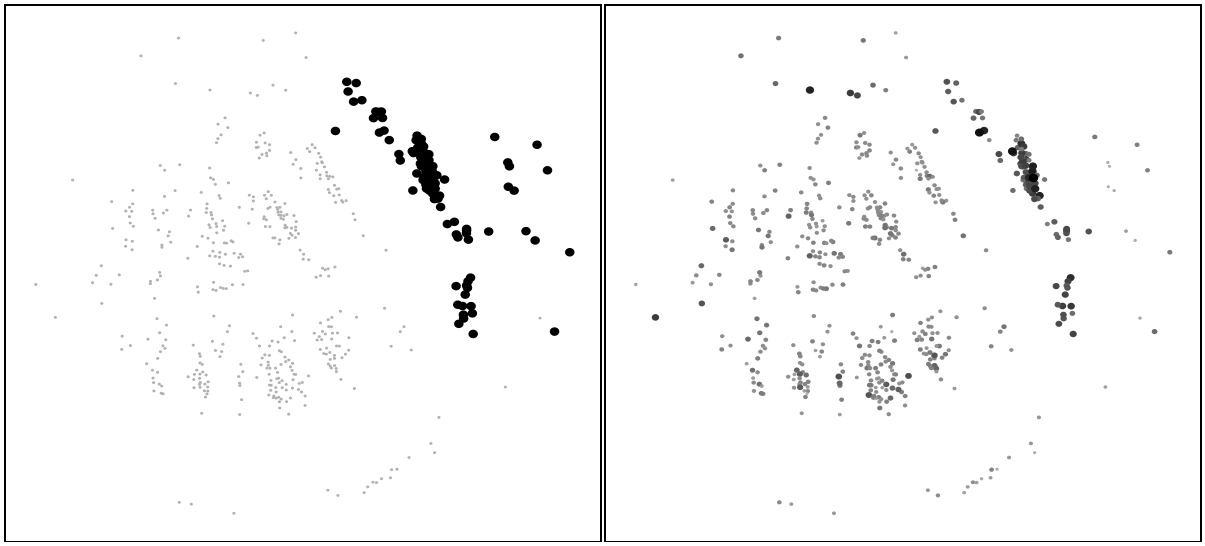
## 4 Summary

We have made numerous experiments with the developed method and different data sets. The results show that visualisation has wide capabilities in the area of data mining. Visual analysis much better uses capabilities of a human being and very frequently leads to the conclusion which are very difficult to access on the basis on rough data. The method and application were dedicated to medical data analysis, but it is much universal—it is possible to use it successfully with other data sets.



(a) P8.2 (b) P10

**Fig. 3.** The geometric separation (data set *cervix*, attributes $P8.2$ i $P10$)



(a) P7.3 (b) P11

**Fig. 4.** The geometric separation (data set *cervix*, attributes $P7.3$ and $P11$)

## References


1. De Backer S., Naud A., and Scheunders P.: *Nonlinear dimensionality reduction techniques for unsupervised feature extraction,* Pattern Recognition Letters, **19,** pp. 711–720, (1998).
2. Dembélé Doulaye and Kastner Philippe: *Fuzzy cmeans for clustering microarray data,* Bioinformatics, **19,** pp. 973–980, (2003).
3. Friendly Michael: *Statistical graphics for multivariate data,* SAS SUGI 16 Conference, (1991).
4. Jain Anil K. and Mao Jianchang: *Artificial neural network of nonlinear projection of multivariate data,* In IEEE International Joint Conference on Neural Networks (6th IJCNN'92), volume III, pages III–335–III–340, Baltimore, MD. IEEE/INNS. MI State U. (1992).
5. König Andreas: *A survey of methods for multivariate data projection, visualisation and interactive analysis,* In Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems, pages 55–59. (1998).
6. Lee R. C. T., Slagle J. R., and Blum H.: *A triangulation method for the sequential mapping of points from Nspace to twospace,* IEEE Transactions on Computers, **C26,** pp. 288–292, (1977).
7. McLeod A. I. and Provost S. B.: *Multivariate data visualization,* In ElShaarawi A. H. and Piegorsch W. W., editors, Encyclopedia of Environmetrics, pages 1333–1344. John Wiley and Sons, (2001).

8. Pekalska E., de Ridder D., Duin R. P. W., and Kraaijveld M. A.: *A new method of generalizing sammon mapping with application to algorithm speedup,* In 5th Annual Conference of the Advanced School for Computing and Imaging, ASCI'99, pages 221–228, (1999).
9. Sammon John W.: *A nonlinear mapping for data stucture analysis,* IEEE Computer, **C18,** pp. 401–409, (1969).

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.159-167
ISBN 83-922646-0-6


# Odkrywanie reguł asocjacji z medycznych baz danych – podejście klasyczne i ewolucyjne

Halina Kwaśnicka, Kajetan Świtalski

Wroclaw University of Technology, Institute of Applied Informatics,
Wyb. Wyspiańskiego 27, 50-370 Wrocław
halina.kwasnicka@pwr.wroc.pl, http://www.ci.pwr.wroc.pl/~kwasnick

**Streszczenie.** W pracy omówiono problematykę generowania reguł asocjacji z medycznych baz danych z wykorzystaniem autorskiej metody generowania reguł asocjacji wykorzystującej algorytm genetyczny EGAR, w porównaniu z klasycznym algorytmem FPTree. Dla celów badawczych opracowano program komputerowy, który jest stosunkowo uniwersalnym narzędziem do tego zadania. W pracy porównano efekty obu metod wykorzystując rzeczywiste zbiory danych medycznych z wrocławskiej kliniki.

## 1 Wprowadzenie

W posiadaniu różnych instytucji znajdują się obszerne zbiory danych, zgromadzone w postaci baz danych bądź hurtowni. Dane te ukrywają często obszerną i bardzo ważną wiedzę, której jednak nie można odczytać z nich bezpośrednio z użyciem standardowych środków. Zrodziła się więc potrzeba zastosowania wydajniejszych rozwiązań, dających lepsze rezultaty. Stąd też pojawienie się nowej dyscypliny naukowej nazywanej dzisiaj drążeniem danych (ang. Data Mining), a będącej częścią szerszego procesu – odkrywania wiedzy z baz danych (ang. Knowledge Discovering from Databases) [2, 10].

Obecnie ze specjalizowanych algorytmów drążenia danych i całego procesu pozyskiwania wiedzy z baz danych korzysta się w takich dziedzinach nauki jak: bankowość, marketing, telekomunikacja, energetyka, farmaceutyka, medycyna i wiele innych. We wszystkich dziedzinach można w zasadzie stosować te same metody drążenia danych, choć należy zwracać uwagę na specyfikę analizowanych danych – w niektórych zastosowaniach posiadane dane zawierają atrybuty zarówno o wartościach ciągłych, jak i symbolicznych, co należy uwzględniać w doborze metod. Niektóre metody nie nadają się do danych niepełnych, zaszumionych, itp., dlatego rodzaj danych jest równie istotny jak i cel drążenia danych. Zawsze w procesie drążenia danych powinni brać udział specjaliści z danej dziedziny, aby wydobyta wiedza mogła być oceniona i zweryfikowana. Czasami eksperci są w stanie postawić interesujące hipotezy, które metody drążenia danych potrafią zweryfikować.

Podstawowy cel tej pracy to zbadanie możliwości generowania reguł asocjacyjnych z medycznych baz danych za pomocą algorytmów genetycznych oraz porównanie użyteczności tej metody z podejściem klasycznym. Zaproponowano genetyczną, autorską metodę EGAR (Extended Genetic Association Rules). Słowo *Extended* oznacza, że autorzy oparli się na metodzie GAR [11]. Opracowany program komputerowy został tak zaprojektowany, aby praca z nim była łatwa i aby wyniki były prezentowane w czytelnej formie. Autorzy pracowali wcześniej z obiema wykorzystanymi tu bazami danych, realizując zadanie klasyfikacji [7, 8].

## 2 Generowanie reguł asocjacji jako zadanie drążenia danych

Za [15] możemy powiedzieć, że: Pozyskiwanie wiedzy z danych jest nietrywialnym procesem wyszukiwania rzeczywistych, nowych, potencjalnie użytecznych i zrozumiałych dla człowieka wzorców w zbiorach danych. Na proces pozyskiwania wiedzy z danych składa się sekwencja zadań, które powinny zapewnić pozyskanie użytecznej i zrozumiałej dla człowieka wiedzy:

1. Zrozumienie dziedziny problemu, zdefiniowanie celu pozyskiwania wiedzy
2. Pozyskanie potrzebnych danych
3. Wstępne przetworzenie danych – konsolidacja i oczyszczenie danych
4. Selekcja odpowiednich danych (redukcja danych) i ich wzbogacenie

5. Zakodowanie danych
6. Drążenie danych (wybór odpowiedniego zadania procesu drążenia, wybór odpowiedniej reprezentacji wiedzy i algorytmu eksploracji danych, uruchomienie algorytmu – przeszukiwanie danych i generowanie znalezionych wzorców)
7. Interpretacja, prezentacja i wyjaśnianie odkrytej wiedzy
8. Konsolidacja i praktyczne wykorzystywanie odkrytej wiedzy

Zadania drążenia danych można podzielić ze względu na postać otrzymywanej wiedzy lub na cel jej wykorzystania [15]. Najpopularniejsze zadania to: *klasyfikacja*, *klasteryzacja* danych, odkrywanie *zależności przyczynowych* oraz odkrywanie *reguł asocjacji* (*reguł związków*). Każde z tych zadań może być realizowane stosując różne algorytmy [5, 13]. Niniejsza praca dotyczy odkrywania reguł asocjacji.

Odkrywanie reguł asocjacji jest najogólniejszym mechanizmem w dziedzinie drążenia danych i może być stosowane również w klasyfikacji przy pewnych założeniach. Typowa postać reguł asocjacji przypomina logiczną implikację:

**Ciało** → **Głowa** [*wsparcie*, *pewność*],

gdzie: **Ciało**, **Głowa** – zbiory atrybutów wraz z odpowiadającymi im wartościami, wsparcie (ang. support) – wsparcie reguły, pewność (ang. confidence) – pewność (ufność) reguły.

*Wsparcie* reguły jest to częstość współwystępowania wartości pewnych atrybutów w bazie danych; podawana jest procentowo. Jeśli wsparcie pewnej reguły wynosi 50%, oznacza to, że w połowie wszystkich danych z bazy, atrybuty występują z wartościami określonymi w ciele reguły. *Pewność* reguły jest to częstość współwystępowania wartości atrybutów z głowy reguły w tych rekordach bazy danych, w których występują wartości atrybutów z ciała reguły. Jeśli pewność reguły wynosi 50%, oznacza to, że w połowie rekordów, w których występują wartości atrybutów z ciała reguły, występują również wartości atrybutów z głowy reguły.

## 3 Metody generowania reguł asocjacji

Można wyodrębnić dwie zasadnicze grupy algorytmów odkrywania reguł asocjacji: algorytmy klasyczne oraz grupa algorytmów obliczeń miękkich (ang. Soft Computing). Algorytmy klasyczne są stosunkowo dobrze poznane, opisane i wykorzystywane w wielu systemach, natomiast algorytmy typu miękkiego stale się rozwijają i nie ma obecnie jednoznacznej metody pozwalającej na osiągnięcie najlepszych wyników. Najbardziej obiecującym podejściem typu *soft computing* wydają się być algorytmy genetyczne i to podejście jest wykorzystane w tej pracy.

### 3.1. Podejście klasyczne

Najprostszy algorytm pozyskiwania reguł to algorytm o nazwie *Apriori* [5]. Jest to algorytm generowania wzorców częstych, na podstawie których można w następnej kolejności wygenerować reguły asocjacji. Jest to iteracyjne podejście, w którym wzorce *k*-elementowe służą do pozyskania wzorców (*k*+1)-elementowych. Wykorzystuje on podstawową własność wzorców częstych: każdy niepusty podzbiór zbioru częstego jest również zbiorem częstym, gdzie poprzez *zbiór częsty* (nazywany też *wzorcem częstym*) rozumiemy zbiór par <atrybut, wartość> współwystępujących w bazie z określoną częstością. Oznacza to, że jeżeli podzbiór pewnego zbioru nie jest częsty, to ten zbiór również nie jest częsty, zatem bezcelowe jest przeglądanie zbiorów będących nadzbiorami zbiorów nieczęstych, w poszukiwaniu wzorców częstych. Pozwala to zawężać przestrzeń przeszukiwań poprzez generowanie „kandydatów" na wzorce częste (*k*+1)-elementowe spośród nadzbiorów zbiorów częstych *k*-elementowych. Przy większej ilości danych wydajność Apriori jest niezadowalająca. Zaproponowano wiele jego modyfikacji mających na celu zwiększenie jego efektywności, jak np.: haszowanie, redukcję skanowania, itp. [5].

*Algorytm drzewa wzorców częstych*, zwany w skrócie FPTree (ang. *Frequent Pattern Tree*) jest algorytmem wydajnego generowania wzorców częstych [10]. Jego istotą jest struktura nazwana drzewem wzorców częstych (FPTree), która pozwala na kompresję danych zawartych w bazie.

Zadaniem algorytmu jest nie tylko stworzenie drzewa, ale też wydobycie tej informacji w postaci zrozumiałej dla użytkownika. Proces ten nazywany jest *drążeniem drzewa*, jest to algorytm rekurencyjny [10].

FPTree jest jednym z najwydajniejszych algorytmów klasycznych. Charakteryzuje się kompletnością (znajdowane są wszystkie wzorce o określonej częstości), jednokrotnym przeglądaniem danych, a co za tym

idzie – dużą wydajnością, brakiem konieczności generowania kandydatów, dużą kondensacją reprezentacji danych (dzięki strukturze FPTree), zachowaniem malenia częstości wzorców. Istnieje wiele wersji algorytmu FPTree, różniących się między sobą głównie zastosowaną optymalizacją działania.

```
function drąż (drzewo, wsparcie)
begin
if ( ma_jedną_gałąź(drzewo) ) generuj_wszystkie_wzorce_z(drzewo)
                    //generowane wzorce to wszystkie kombinacje elementów jedynej gałęzi
else if drzewo jest puste=return {}
     for_each(elementy e drzewa) do
          make wzorce_warunkowe for element e
                    //znajduje scieżki zawierające e
          filter wzorce_warunkowe, wsparcie
                    //pozostawia jedynie elementy wystarczającym wsparciu
          make drzewo_warunkowe z wzorce_warunkowe
                    //podczas budowy drzewa warunkowego wzorce warunkowe są
                    // traktowane tak, jak rekordy podczas budowy drzewa głównego
          nowe_wzorce = drąż (drzewo_warunkowe, wsparcie)
          for_each (nowe_wzorce) add element e
          result += nowe_wzorce
     end_for_each
return result
end
```

**Rysunek 1. Pseudokod zaimplementowanego algorytmu FPTree**

W pracy zastosowano prosty algorytm służący generowaniu reguł asocjacji na podstawie otrzymanych uprzednio wzorców częstych. Algorytm ten polega na wygenerowaniu dla każdego wzorca częstego **W** wszystkich kombinacji reguł typu: $\mathbf{A} \Rightarrow \mathbf{B}$, gdzie **A** to dowolny podzbiór zbioru częstego, a **B** to różnica zbiorów **W\A**. Wsparcie tak utworzonej reguły równe jest częstości zbioru częstego, z którego została utworzona, natomiast pewność jest ilorazem wsparcia zbioru częstego **W** i wsparcia zbioru **A**.

FPTree można stosować jedynie do atrybutów dyskretnych, w systemie opracowanym na potrzeby niniejszej pracy do dyskretyzacji atrybutów o dziedzinach ciągłych wykorzystano *algorytm dyskretyzacji według równej częstości*. Jest to prosty algorytm należący do grupy „bez nadzoru", polega na dzieleniu zakresu wartości oryginalnego atrybutu na ustaloną z góry liczbę przedziałów. Granice tych przedziałów są dobierane tak, aby możliwie w każdym z nich znalazła się ta sama liczba przykładów uczących.

### 3.2. Algorytm genetyczny jako narzędzie drążenia danych (soft computing)

Algorytm genetyczny jest wzorowaną na naturalnej ewolucji metodą optymalizacyjną, nie gwarantującą znalezienia optymalnego rozwiązania [4, 6]. Jest jednak silną, uznaną metodą przeszukiwania, dającą często zadowalające wyniki.

Idea algorytmów genetycznych polega na zakodowaniu potencjalnego rozwiązania, utworzeniu populacji początkowych rozwiązań (nawet losowo), a następnie ich 'ewoluowaniu' stosując mechanizmy zaczerpnięte z biologii: lepsze osobniki są statystycznie częściej reprodukowane (mają więcej 'dzieci'), pokolenie potomne ('dzieci') różni się od swoich rodziców wskutek losowego działania mechanizmów różnicujących – mutacji i krzyżowania. Po pewnym czasie ewolucji, przy dobrze dobranych parametrach ewolucji i ocenie osobników, otrzymujemy satysfakcjonujące (czasami optymalne) rozwiązania. Dzięki swojej sile, algorytmy genetyczne znalazły szerokie zastosowanie w rozwiązywaniu problemów szeregowania zadań, modelowania finansowego, optymalizacji funkcji, harmonogramowania, itp. Znajdują też szerokie zastosowanie w analizie danych medycznych, w tym, w drążeniu danych zgromadzonych w różnych klinikach świata [6, 7, 14].

Zaproponowany w niniejszej pracy algorytm genetyczny wykorzystuje doświadczenia autorów systemu GAR (Genetic Association Rules), który daje interesujące wyniki dla danych zawierających atrybuty z ciągłymi wartościami [11]. GAR nie generuje bezpośrednio reguł asocjacji, ale generuje wzorce częste, z których dopiero można wygenerować reguły asocjacji. Wiele baz danych, jak np. medyczne bazy danych, zawiera atrybuty zarówno o wartościach ciągłych jak i dyskretnych (symbolicznych), stąd nazwa zaproponowanej modyfikacji – EGAR (*Extended GAR*). Modyfikacji uległ zarówno sposób reprezentacji osobnika, jak i operatory genetyczne – mutacja i krzyżowanie. Tak przystosowany algorytm jest bardziej uniwersalny, może mieć szersze zastosowanie.

EGAR to podejście typu *Mitchigan* [4, 6] a więc genotyp osobnika zawiera informację o cząstkowym rozwiązaniu problemu – pojedynczym wzorcu częstym. Drążenie większej liczby wzorców polega na wielokrotnym przeprowadzaniu procesu ewolucji, przy czym każdorazowo z ostatniego pokolenia populacji wybierany jest najlepiej przystosowany osobnik, a reprezentowany przez niego wzorzec częsty dopisany zostaje do globalnego zbioru wzorców częstych. Liczba wzorców pozyskanych z bazy zależna jest więc wyłącznie od liczby iteracji algorytmu.

Genotyp osobnika składa się z dwóch chromosomów, w którym zapisane są informacje o atrybutach występujących w zbiorze częstym oraz przedziałach, do których należą ich wartości. Jeden chromosom zawiera atrybuty ciągłe, gen w tym chromosomie jest trójką <a,*l*,*u*>, gdzie *a* – atrybut, *l* – minimalna wartość (*lower value*), *u* – *maksymalna wartość* (*upper value*). Drugi chromosom tego samego osobnika zawiera atrybuty o dyskretnych wartościach, gen jest tu dwójką <*a*,*v*>, gdzie *a* – atrybut, *v* – wartość atrybutu. Długość chromosomów może być różna u poszczególnych osobników, zależy od aktualnej liczby atrybutów (dyskretnych i ciągłych) w reprezentowanym przez osobnika wzorcu częstym.

Doświadczenie wskazuje, że w zadaniach drążenia danych lepiej sprawdzają się twarde metody selekcji, to znaczy takie, które w większym stopniu preferują najlepszych osobników. W prezentowanym podejściu zastosowano metodę próbkowania deterministycznego z elitaryzmem [4, 6]. Elitaryzm polega na preferowaniu najlepszych osobników, w tym wypadku, najlepszy osobnik z każdego pokolenia przechodzi bez zmian do następnego pokolenia (tzw. przeżywanie najlepszego). Jest to szczególnie istotne, biorąc pod uwagę fakt, że właśnie najlepiej oceniony osobnik dołączony zostaje do globalnego zbioru wzorców częstych.

Bardzo ważnym elementem jest funkcja oceny osobnika, tzn., na ile rozwiązanie zakodowane przez danego osobnika dobrze spełnia zadanie. W zadaniu generowania reguł asocjacji ocena ta jest wielokryterialna. Najbardziej naturalnym sposobem wydaje się być liczba pokrytych rekordów w bazie przez danego osobnika (kryterium *trafności*), ale wtedy preferowane byłyby osobniki trywialne, jedno-atrybutowe [8]. Dlatego też funkcja oceniająca jest uzależniona od innych miar, np. długości chromosomu (czyli liczby atrybutów występujących we wzorcu). Aby powiększyć miarę trafności generowanych wzorców ewolucja ma tendencję do poszerzania szerokości przedziałów wartości dla poszczególnych atrybutów aż do objęcia całej dziedziny atrybutu jednym przedziałem. Jest to oczywiście niepożądana cecha ewolucji, dlatego też w ocenie osobnika należy karać go za zbyt dużą *średnią amplitudę* przedziałów, gdzie średnia amplituda to średnia długość przedziałów zawartych w genach osobnika. Ta cześć oceny dotyczy chromosomu zawierającego atrybuty ciągłe. Aby nie generować wzorców częstych pokrywających te same rekordy w bazie, do funkcji oceny osobników włączono karę za pokrywanie już pokrytych rekordów. W tym celu, w każdej iteracji algorytmu, kiedy zwycięski wzorzec (osobnik) dopisywany jest do zbioru wzorców częstych, pokrywane przez niego rekordy są oznaczane. Podsumowując, funkcja oceny (przystosowanie) *i*-tego osobnika wyraża się wzorem:

$$f_i = \mathrm{cov}_i - a \cdot mark_i - b \cdot ampl_i + c \cdot nAtr_i$$

gdzie: $cov_i$ – liczba rekordów pokrytych przez *i*-ty wzorzec, $mark_i$ – liczba takich rekordów pokrytych przez *i*-ty wzorzec rekordów, które były pokryte przez wcześniejsze wzorce, $ampl_i$ – średnia amplituda wzorca, $nAtr_i$ – liczba atrybutów w kodowanym wzorcu, *a*, *b*, *c* – wagi kar i nagród ustalane przez użytkownika.

Funkcja przystosowania zależna jest od obiektywnych miar jakości osobnika oraz od parametrów warunkujących wpływ tych miar, dlatego też EGAR wymaga dostrojenia wag przez użytkownika.

Zdefiniowano specjalizowane operatory dla poszczególnych chromosomów. Krzyżowanie osobników rodzicielskich daje dwóch potomków, z których lepszy (o większym przystosowaniu) jest wybierany do następnego pokolenia. Chromosomy zawierające atrybuty ciągłe są krzyżowane jak w GAR: chromosom pierwszego z potomków powstaje poprzez przepisanie wszystkich atrybutów z pierwszego kojarzonego osobnika, przy czym zakresy przedziałów wartości poszczególnych atrybutów sumowane są z odpowiednimi zakresami z chromosomu drugiego osobnika biorącego udział w krzyżowaniu, o ile w tym drugim atrybuty takie występują. Jeśli w chromosomie drugiego z krzyżowanych osobników nie występuje dany atrybut, to wartości krańców przedziału nie ulegają zmianie, to znaczy są takie same jak w pierwszym z kojarzonych osobników. Drugi chromosom potomny powstaje zaś poprzez przepisanie genów z drugiego chromosomu pierwszego osobnika biorącego udział w krzyżowaniu, a następnie dodanie wybranych w sposób losowy genów z drugiego chromosomu drugiego kojarzonego osobnika. W przypadku, gdy wylosowany do dodania gen reprezentuje atrybut istniejący już w tworzonym chromosomie, jego wartość zostaje zastąpiona nową z prawdopodobieństwem 50%. Analogicznie powstaje chromosom drugiego potomka, przy czym kojarzone osobniki zamieniane są miejscami.

*Mutacja chromosomu z atrybutami ciągłymi* polega na zmianie jednego lub większej liczby genów, poprzez modyfikację zawartych w nich krańców przedziałów. Wyróżnia się cztery możliwości mutacji genu: przesunięcie całego przedziału w lewo lub prawo, oraz zmniejszenie lub zwiększenie przedziału. Prawdopodobieństwo mutacji ustala się w granicach 1%, przy czym najczęściej przeprowadzana jest mutacja pojedynczego genu. Chromosom z atrybutami dyskretnymi podlega dwóm rodzajom mutacji: pierwsza polega na zastąpieniu dys-

kretnej wartości atrybutu inną, wskazaną na podstawie wartości tego atrybutu w losowo wybranym rekordzie bazy danych. Ten sposób mutacji uwzględnia rozkład wartości danego atrybutu w bazie danych, ponieważ większe jest prawdopodobieństwo wylosowania wartości atrybutu częściej występującego w bazie. Jest to istotne, bo zwiększa szanse na odpowiednie wsparcie osobnika podlegającego mutacji. Drugi rodzaj mutacji polega na zastąpieniu atrybutu reprezentowanego przez dany gen innym, losowo wybranym, dyskretnym atrybutem bazy oraz przypisaniu mu losowo wybranej wartości, przy czym w tym przypadku nowa wartość atrybutu jest losowana wprost z jego dziedziny. W ten sposób, zapewnia się, że każda wartość należąca do dziedziny atrybutu ma tę samą szansę na wylosowanie, dzięki czemu możliwe jest również odkrycie wzorców o mniejszym wsparciu.

## 4 Eksperymenty na medycznych bazach danych

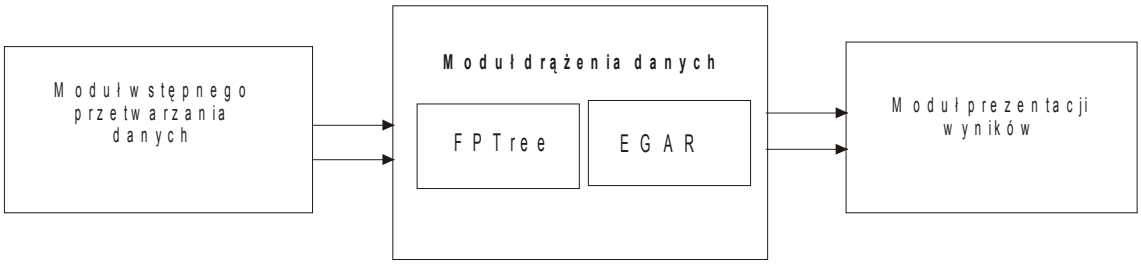
Do celów eksperymentalnych zaprojektowano i zaimplementowano obie omówione wcześniej metody generowania reguł asocjacyjnych na bazie wzorców częstych: FPTree i EGAR. Wykonany program komputerowy jest na tyle uniwersalny, że umożliwia badania na różnych bazach danych zawierających atrybuty zarówno dyskretne, jak i ciągłe.

### 4.1 System Antlia

System Antlia (nazwa wynika z zamiłowania astronomią współautora badań, Antlia jest nazwą gromady) jest wygodnym narzędziem do generowania reguł asocjacyjnych z danych zawierających zarówno atrybuty o wartościach ciągłych, jak i dyskretnych. Pod względem logicznym program zbudowany jest z trzech zasadniczych części: *modułu wstępnego przetwarzania danych*, *modułu drążenia danych* oraz *modułu prezentacji wyników*. Każda z tych części reprezentowana jest poprzez jedną z ikon w górnej części okna systemu. Użytkownik ma w każdej chwili możliwość przełączenia pomiędzy tymi modułami. *Antlia* umożliwia przeprowadzenie następujących operacji:

- wczytanie, wstępne odfiltrowanie oraz sortowanie danych,
- dyskretyzację danych metodą stałej częstości,
- definiowanie przez użytkownika liczby punktów podziału dyskretyzowanych dziedzin,
- zapis wstępnie przetworzonych danych w nowym pliku,
- drążenie reguł asocjacji metodą *FPTree* oraz autorską metodą *EGAR*,
- ustalanie wszystkich parametrów wybranej metody drążenia,
- śledzenie na wykresie wpływu wybranych parametrów na średnie przystosowanie populacji w algorytmie *EGAR*,
- przeglądanie, sortowanie oraz filtrowanie otrzymanych reguł asocjacji,
- zapisywanie reguł asocjacji do pliku, ich wydruk oraz podgląd wydruku.

Na rys. 2. przedstawiono ogólny schemat blokowy prezentowanego systemu. Program Antlia został napisany w środowisku Microsoft Visual Studio .NET 2002, w języku C++. W celu zapewnienia odpowiedniej wydajności, wszystkie algorytmy wykorzystywane w systemie zostały zoptymalizowane przy użyciu profilera, będącego częścią oprogramowania Compuware DevPartner Studio. Kod programu został obdarzony odpowiednim komentarzem w języku angielskim, według standardów DoxyGene.



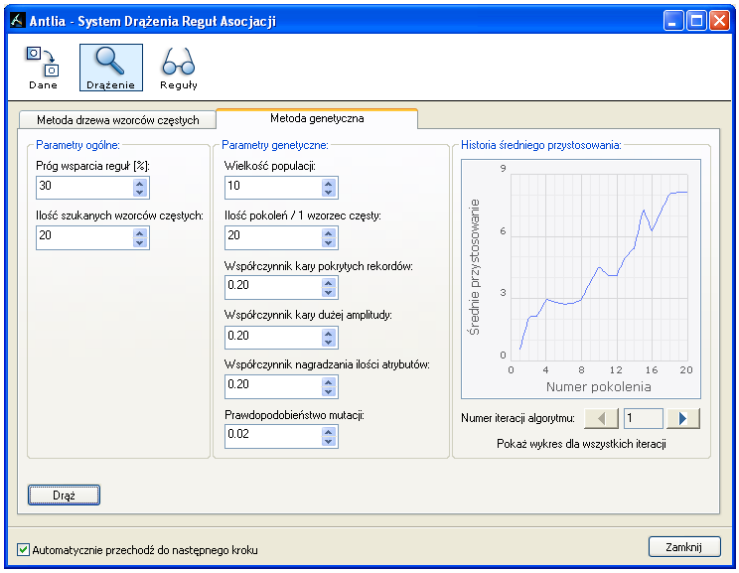

**Rysunek 2. Podział systemu Antlia na moduły logiczne**

*Moduł wstępnego przetwarzania danych* odpowiedzialny jest za odczytanie danych ze wskazanego pliku oraz ich wstępne przygotowanie do drążenia wybraną metodą. Program obsługuje własny format plików, przy-

gotowanie danych w tym formacie jest bardzo proste, sprowadza się do wyeksportowania danych w postaci pliku tekstowego, który musi spełniać dwa warunki: (1) w pierwszej linii pliku powinny znajdować się nazwy atrybutów oddzielone znakami tabulacji (nazwy mogą zawierać inne znaki białe np. spacje), (2) w każdej następnej linii pliku powinny znaleźć się wartości atrybutów oddzielone znakami tabulacji. Wczytane dane widoczne są w oknie programu w postaci tabeli. Dane te można sortować rosnąco lub malejąco według dowolnego atrybutu, możliwe jest również odfiltrowanie części rekordów. Ta część systemu umożliwia dyskretyzację atrybutów ciągłych, szczególnie ważną w przypadku drążenia reguł asocjacji metodą *FPTree*. Dyskretyzacja przeprowadzana jest *według stałej częstości*, przy czym liczba punktów podziału przeciwdziedziny atrybutu może być zdefiniowana przez użytkownika. Dodatkową możliwością jest wybór zakresu dyskretyzacji, daje to możliwość dyskretyzacji wszystkich atrybutów numerycznych, lub jedynie atrybutów o dziedzinach rzeczywistych.

*Moduł drążenia danych* daje użytkownikowi możliwość wyboru metody drążenia danych, dobrania jej parametrów, a następnie przeprowadzenia procesu drążenia reguł asocjacji. Moduł ten składa się z dwóch części, z których każda odpowiedzialna jest za osobną metodę drążenia danych. Użytkownik wybiera odpowiednią metodę drążenia danych poprzez wybór odpowiedniej zakładki. Dla metody *drzewa wzorców częstych*, głównymi parametrami ustalanymi przez użytkownika są: minimalne wsparcie oraz minimalna pewność szukanych reguł. Dodatkową możliwością oferowaną przez system jest wybór atrybutów, które, ze względu na cel drążenia danych, są interesujące w części konkluzji reguły. Opcja ta daje również możliwość przeprowadzenia procesu klasteryzacji danych, przy czym wartość atrybutu wybranego do części konkluzji (głowy reguły) określa przynależność do klasy, obiektu reprezentowanego przez rekord bazy danych.

Wybór drugiej zakładki w module drążenia danych powoduje przejście do *metody* EGAR. Dostępne tu pola umożliwiają ustalenie wszystkich parametrów algorytmu EGAR i przeprowadzenie drążenia danych w poszukiwaniu reguł asocjacji. Parametry konfigurujące algorytmu EGAR podzielone są na parametry ogólne, jak: minimalne wsparcie i liczba szukanych wzorców częstych oraz parametry specyficzne dla algorytmu genetycznego, czyli: wielkość populacji, liczba pokoleń przypadająca na wzorzec częsty, współczynnik kary dla pokrytych rekordów, współczynnik kary dla dużej amplitudy, współczynnik nagradzania liczby pokrytych rekordów oraz prawdopodobieństwo mutacji. Użytkownik ma możliwość śledzenia wpływu parametrów na działanie algorytmu, poprzez wykres średniego przystosowania populacji na przestrzeni pokoleń (rys. 3.).




**Rysunek 3. Ekran ustalania parametrów metody EGAR**

Ostatnia część systemu Antlia, to moduł *prezentacji wyników*, służy do przeglądania i weryfikacji otrzymanych reguł asocjacji. Funkcje dostępne w tej części systemu umożliwiają użytkownikowi również sortowanie i filtrowanie reguł według osiąganego przez nie wsparcia i pewności. Możliwe jest zapisanie odfiltrowanych reguł asocjacji w wybranym przez użytkownika pliku.

### 4.2. Charakterystyka baz danych

Do eksperymentów wykorzystano medyczne bazy danych *sutek.xls* (dotyczy chorych kobiet na raka sutka) i *szyjka.xls* (dotyczy raka szyjki macicy). Obie bazy były przedmiotem wcześniejszych badań dla zadania klasyfikacji, z wykorzystaniem metod: analiza statystyczna [12], ewolucyjne generowanie reguł klasyfikujących [7] oraz wizualizacja danych [8]. W tych badaniach współpracujący lekarze wskazali cel badań i analizowali wyniki.

Wykorzystane bazy danych, jak większość rzeczywistych medycznych baz danych, nie są najłatwiejsze dla automatycznego pozyskiwania wiedzy, są stosunkowo małe, zawierają brakujące dane, zawierają atrybuty różnych typów: symboliczne, numeryczne – dyskretne i ciągłe. Większość danych opisuje tzw. typowe przypadki, co sprzyja generowaniu wiedzy typowej, znanej lekarzom. Baza *sutek.xls* zawiera 100 rekordów, dane dotyczą 21 atrybutów o różnych dziedzinach wartości. Baza *szyjka.xls* zawiera 530 rekordów, 12 atrybutów o różnych dziedzinach wartości. Należy podkreślić, że brak danych dla takich atrybutów, jak *wiek zgonu* jest oczywisty dla pacjentek, które jeszcze żyją, ale brak samej informacji o ponad 30 pacjentkach czy *żyją* jest większym problem.

### 4.3 Analiza wyników

Na początku przeprowadzono drążenie danych klasyczną metodą FPTree, która pozwala na wygenerowanie wszystkich reguł asocjacji o zadanych parametrach wsparcia i pewności.

***Eksperymenty z metodą FPTree***: Pierwsze eksperymenty odbyły się z bazą sutek.xls, bez dyskretyzacji i filtrowania danych, wsparcie i pewność ustalono na 70%. Otrzymano 340 reguł asocjacji o średnim wsparciu 73% i pewności 93%, ale – zgodnie z oczekiwaniami autorów, większość reguł była pewna, ale trywialna, np. jeśli nie nastąpił nawrót choroby, to pacjentka żyje lub, jeśli w wywiadzie krewnych nie stwierdzono raka sutka, to nie stwierdzono też również innego raka (92% pewność). W regułach, z powodu braku dyskretyzacji, nie występowały atrybuty o wartościach rzeczywistych, szukaliśmy reguł o wysokim wsparciu (70%), można oczekiwać więc, że tak wysokie wsparcie w medycznych bazach danych mają typowe przypadki, traktowane rutynowo przez lekarzy. Zatem, następne eksperymenty wykonano przy obniżonym wsparciu do 50%, ale – aby uzyskana wiedza nie była przypadkowa, podniesiono próg pewności do 95%.

Analizowano reguły po odfiltrowaniu tych ze wsparciem powyżej 70% (algorytm jest deterministyczny, te reguły mieliśmy w poprzednim eksperymencie), było ich 355, ich średnie wsparcie 56% a średnia pewność – aż 99%. Otrzymane reguły dotyczą w większości zależności okresu przeżycia pacjentek oraz czasu nawrotu choroby od stopnia złośliwości histopatologicznej. Dla laika jest to bardzo interesująca wiedza, jednakże dla lekarza nie stanowi ona wiedzy odkrywczej. Dlatego w kolejnym eksperymencie wskazano atrybuty, które mają znaleźć się w części konkluzji generowanych reguł asocjacji. Zgodnie z zainteresowaniem lekarzy w poprzednich badaniach, do konkluzji włączono atrybut *długość okresu przeżycia* oraz *czas do wystąpienia nawrotu*. Przy minimalnych wsparciu i pewności równych 70% uzyskano 56 reguł o średnim wsparciu 74% i pewności 96%. Jakkolwiek uzyskane reguły są w większości trywialne, zdarzają się też bardziej interesujące. Przykładowo, jeśli u bliskich krewnych nie było raka sutka oraz pacjentka w przeszłości nie chorowała na raka, to w większości przypadków (86%) ma czas przeżycia większy od 5 lat (maksymalny wyróżniony okres w danych).

Szczegółowa analiza wyników wskazuje, że w regułach występuje tylko maksymalny okres przeżycia, ponieważ występuje on najczęściej, a reguły o innej wartości nie są w stanie uzyskać wystarczającego wsparcia. Można w tej sytuacji odfiltrować z danych wstępnie te rekordy, które mają maksymalną wartość interesujących nas atrybutów lub znacznie obniżyć wymagany próg wsparcia. Pierwsze rozwiązanie wiąże się ze zbytnim zubożeniem zbioru danych – zbiór stanie się mało liczny, w drugim – będzie bardzo dużo częstych wzorców i czas drążenia znacznie się wydłuży, uzyskana liczba reguł będzie zbyt duża, aby je móc przeanalizować. Zastosowano 'ręczną' dyskretyzację danych numerycznych, dla *długość przeżycia* i *czas nawrotu* wyodrębniono dwa przedziały, jednowartościowy o największej wartości i drugi, obejmujący pozostałe wartości. Uzyskane reguły osiągnęły 100 % pewność i wsparcie blisko 16%. Poza trywialnymi regułami uzyskano też interesujące, np. jeśli nastąpi wznowa z odległymi przerzutami to okres przeżycia jest mniejszy niż 5 lat.

Aby sprawdzić wpływ dyskretyzacji na generowanie reguł asocjacji metodą FPTree dokonano dyskretyzacji wszystkich danych ciągłych w bazie. Obniżono parametr wsparcia do 30%, bo po dyskretyzacji wartości ciągłych, uzyskanie wysokiego wsparcia nie jest możliwe (automatyczne podzielenie przedziału wartości na $n$ zbiorów powoduje, że maksymalne wsparcie może wynosić $100/n$ procent). Uzyskane reguły zawierały atrybuty rzeczywiste, ich średnie wsparcie to 39% a pewność 88%. Reguła, którą należałoby dać lekarzom do interpretacji to: jeśli stopień zawansowania klinicznego wg UICC jest 2b oraz wielkość guza należy do przedziału [3,5 cm – 7,5 cm] to w wywiadzie u bliskich krewnych nie stwierdzono raka. Dyskretyzacja

wszystkich danych numerycznych powoduje generowanie dużej liczby reguł, trudno je analizować, ciężko znaleźć wśród nich interesujące.

Wykorzystanie bazy szyjka.xls powodowało te same problemy, co poprzednia baza – generowanie dużej liczby dość znanych i trywialnych reguł. Do ciekawszych reguł można zaliczyć regułę ukazującą zależność pomiędzy miejscem zamieszkania a rodzajem raka: jeśli pacjentka mieszka w mieście, to rodzaj histopatologiczny nowotworu jest *Ca plano* (pewność 91,18%). Atrybuty numeryczne zawierają mało powtarzające się wartości, dlatego też te atrybuty nie występują w odkrywanych regułach, mimo obniżania wartości wsparcia do 25%. Postępując podobnie, jak dla bazy sutek.xls, dokonując dyskretyzacji danych numerycznych i ograniczając listę atrybutów mogących wystąpić w części konkluzji, otrzymano kilka ciekawych reguł, np. jeśli w drugim leczeniu zastosowano izotop promieniotwórczy – rad, a rodzaj histopatologiczny był *Ca plano*, to pacjentka nie przeżyła.

***Eksperymenty z metodą EGAR***: Algorytm ten wymaga ustalania znacznie większej liczby parametrów niż FPTree, co wymaga większego zaangażowania ze strony użytkownika. Jednocześnie, metoda ta zwalnia użytkownika z konieczności dyskretyzacji atrybutów, jako że sama może dobierać granice przedziałów wartości atrybutów ciągłych (numerycznych). Jak zostało wspomniane w opisie programu, pozwala on na podgląd średniego przystosowania w populacji. Informacja ta jest ważna dla użytkownika, bowiem może on śledzić, czy przy dobranych parametrach ewolucja przebiega odpowiednio, tzn., czy nie dochodzi do przedwczesnej zbieżności, albo czy nasz algorytm nie przypomina błądzenia losowego zamiast ukierunkowanych zmian. Autorzy starali się tak dobrać domyślne parametry metody EGAR, aby w większości przypadków możliwe było uzyskanie interesujących reguł. Jak pamiętamy, w tym algorytmie liczba uzyskiwanych wzorców częstych jest parametrem metody, w każdej iteracji znajdowany jest jeden wzorzec częsty. Z punktu widzenia algorytmów genetycznych, istotnymi badaniami są eksperymenty mające na celu pokazanie wrażliwości metody na parametry operatorów genetycznych, wielkość populacji itp. Z uwagi na ograniczone miejsce, nie będziemy przytaczać wszystkich wykonanych eksperymentów. Skupimy się na podsumowaniu tych badań. Warto zwrócić uwagę, że baza sutek.xls zawiera sporo atrybutów numerycznych, co powinno pozwolić na weryfikację zdolności metody do samodzielnego szukania odpowiednich przedziałów wartości tych atrybutów w regułach. Interesowano się głównie regułami o wysokiej pewności i niezbyt wysokim wsparciu, mając nadzieję, że właśnie w tej grupie znajdą się interesujące, tzn. odkrywcze reguły.

Bolączką tej metody jest skłonność do generowania reguł zbyt ogólnych, nie reprezentujących żadnej istotnej wiedzy. Wynika to z faktu, że EGAR, starając się spełnić wymóg dużego wsparcia dla generowanych reguł, jest skłonny ustalać przedziały wartości dla atrybutów numerycznych możliwie szerokie, nawet jeden przedział dla całej dziedziny. Aby temu przeciwdziałać, należy odpowiednio dobrać parametr *współczynnik kary dużej amplitudy*. Zbyt mała jego wartość spowoduje generowanie szerokich przedziałów wartości dla atrybutów numerycznych, zbyt duża wartość może zablokować proces ewolucji. Należy dbać, aby następował wzrost średniego przystosowania populacji w kolejnych iteracjach algorytmu. Zbytnie ujednolicenie się osobników w populacji może być spowodowane za małym prawdopodobieństwem mutacji. Pamiętajmy również o odpowiednim ustaleniu parametru *współczynnik kary pokrytych rekordów*, który nie pozwala na to, aby kolejne generowane reguły pokrywały te same wzorce w bazie danych – reguły te reprezentowałyby tę samą wiedzę. Z istoty algorytmu wynika, iż w pierwszych pokoleniach ewolucji głównie wsparcie i pewność generowanych reguł odgrywa rolę w ewolucji, później stopniowo do głosu dochodzą parametry związane z pokrywaniem wzorców już pokrytych oraz z amplitudą przedziałów dla atrybutów numerycznych. Pozwala to na odkrywanie reguł o stosunkowo dużym wsparciu, które nie były generowane metodą FPTree, np., że pacjentki z zawartością białka *nm*23 nie większą niż 6 przeżywały nie mniej niż 3 lata (92% pewność) lub u pacjentek, których bliscy krewni nie chorowali na raka, czas wznowy był nie krótszy niż 3 lata. Eksperymenty wykazały, że odpowiedni dobór współczynników kar za pokrywanie już pokrytych reguł oraz za amplitudę szerokości przedziałów wartości atrybutów numerycznych pozwala na wyeliminowanie nieefektywnych przebiegów ewolucji, tzn. takich, w których średnie przystosowanie populacji pozostaje na niskim poziomie. Jeśli parametr *nagroda ilości atrybutów* jest zbyt niski, to generowane reguły są stosunkowo krótkie. Z jednej strony jest to pozytywna cecha, bo generowane są reguły dość ogólne, ale z drugiej – takie reguły zwykle nie zawierają interesującej wiedzy. Preferując nieco dłuższe reguły, pozwalając na stosunkowo wysokie kary za dużą amplitudę, możemy uzyskać efektywne ewolucje oraz interesujące reguły, zawierające atrybuty numeryczne.

Przeprowadzono badanie efektywności metody EGAR na bazie sutek, w której atrybuty numeryczne zostały zdyskretyzowane wcześniej (EGAR nie wymaga dyskretyzacji). W wyniku otrzymano tylko 14 reguł, o stosunkowo wysokiej pewności, ale niebezpieczeństwem było utykanie algorytmu w lokalnym optimum – już po kilku pokoleniach średnie przystosowanie populacji przestało wzrastać.

Na obu bazach danych eksperymenty wykazują, że w wzrost prawdopodobieństwa mutacji pomaga zlikwidować przedwczesną zbieżność populacji. Jest to obserwacja zgodna z przesłankami teoretycznymi, to właśnie mutacja wnosi nowe wartości genów do populacji. Na uwagę zasługuje obserwacja (zgodna z teoretyczną ana-

lizą metody), że drążenie reguł z baz dyskretnych jest procesem bardziej losowym z uwagi na fakt, że losowość wprowadzanych podczas mutacji genów odgrywa dużą rolę i przeszkadza to w ukierunkowaniu ewolucji.

## 5. Podsumowanie

Przeprowadzone eksperymenty oraz dokładna analiza metod pozwala stwierdzić, że metoda FPTree jest stosunkowo efektywną metodą dla pozyskiwania reguł asocjacji z baz danych o atrybutach dyskretnych. Możliwe jest stosunkowo szybkie uzyskanie **wszystkich** wzorców częstych, a więc i zawartych w bazie reguł asocjacji, o zadanych wartościach minimalnego poziomu wsparcia i pewności. Stosowanie EGAR w takim przypadku nie gwarantuje znalezienia wszystkich reguł, co przemawia na jego niekorzyść. Zaprojektowany program pozwala na wskazanie atrybutów interesujących w części konkluzji, co pozwala na ukierunkowanie procesu generacji reguł. Jednakże dla drążenia wiedzy z danych numerycznych, zwłaszcza ciągłych, metoda EGAR lepiej się sprawdza. Elastyczność automatycznego doboru przedziałów wartości poszczególnych atrybutów wykazała przewagę nad sztucznym podziałem ich dziedzin na ustaloną z góry liczbę przedziałów, umożliwia to uzyskiwanie reguł asocjacji o wyższym wsparciu. Należy też podkreślić, że EGAR jest metodą wrażliwą na parametry, jej stosowanie wymaga od użytkownika pewnego wysiłku i wyczucia. Możliwość podglądania wykresów przebiegu ewolucji ułatwia odpowiedni dobór parametrów. Biorąc pod uwagę, że mamy do czynienia z medycznymi bazami danych, należy zwrócić uwagę na fakt, że większość rekordów w bazie to bardziej typowe przypadki, dlatego też chcąc znaleźć odkrywcze reguły należy tak dobrać samą metodę pozyskiwania wiedzy, jak i jej parametry, aby nie promować zbytnio wysokiego wsparcia reguł, ponieważ doprowadzi to do reguł reprezentujących wiedzę dość dobrze znaną i stosowaną przez lekarzy. Trzeba uciec się do generowania reguł niezbyt ogólnych, o wysokiej pewności i niezbyt wysokim wsparciu.

Stworzone narzędzie badawcze może służyć do pozyskiwania reguł z baz danych, dając możliwości wyboru samej metody, faktu i sposobu dyskretyzacji danych, filtrowania i sortowania danych wykorzystywanych do drążenia, filtrowania i sortowania pozyskanych reguł, ich wygodnej prezentacji, również w postaci nadającej się do druku. System, poprzez wskazanie atrybutów do konkluzji generowanych reguł może być wykorzystany do generowania reguł klasyfikacji. Planowane jest rozbudowanie systemu o moduł drążenia danych poprzez wizualizację danych, wykorzystując wyniki uzyskane w [8].

## References


1. Aggarwal R., Prasad V.: *A tree projection algorithm for generation of frequentitemsets*. Journal of Parallel and Distributed Computing, 2001.
2. Francisci D., Brisson L., Collard M.: *A Scalar Evolutionnary Approach to Rule Extraction.* Laboratoire Informatique Sinaux et Systemes, 2003.
3. Freitas A.: A *Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery.* Pontificia Universidade Catolica do Parana, 2002.
4. Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.
5. Han J., Kamber M.: *Data Mining: Concepts and Techniques.* Morgan Kauf., 2000.
6. Kwaśnicka Halina: *Obliczenia ewolucyjne w medycynie.* W: Kompendium informatyki medycznej. Red. Radosław Zajdel [i in.]. [Bielsko-Biała]: Alfa Medica Press, corp. 2003 s. 365-402, 2003.
7. Kwaśnicka H., Markowska-Kaczmar U., Matkowski R., Dryl J., Mikołajczyk P., Tomasiak J.: *Rule Discovery from Medical Data Using Genetic Algorithm.* Fourth International ICSC Symposium on Engineering of Intelligent Systems, Portugal, 2004.
8. Kwaśnicka H., Markowska-Kaczmar U., Matkowski R., Dryl J., Mikołajczyk P., Tomasiak J.: *Discovering Dependencies in Medical Data by Visualisation.* International ICSC Symposium on Engineering of Intelligent Systems, Portugal, 2004.
9. Lavington S.H., Freitas A.: *Mining Very Large Databases with Parallel Processing*. Kluwer, 1998.
10. Mao R., Yin Y., Pei P.: *Data Mining and Knowledge Discovery*. Kluwer Academic Publishers, 2004.
11. Mata J., Alvarez J. L., Riquelme J. C.: *An Evolutionary Algorithm to Discover Numeric Association Rules.* Universidad de Huelva, 2001.
12. Matkowski R.: *Wartość prognostyczna ekspresji receptora estrogenowego i produktu genu nm23 w komórkach raka przewodowego. Metody drążenia danych w zastosowaniach medycznych i ich korelacja z wybranymi parametrami klinicznymi.* Akademia Medyczna we Wrocławiu, 2002 .
13. Olivia Parr Rud: *Data Mining Cookbook*. Wiley Computer Publishing, John Wiley & Sons, Inc, 2001
14. Peña-Reyes C. A, Sipper M. *Evolutionary computation in medicine: an overview.* Artificial Intelligence in Medicine 2000; **19** (1):1-23.
15. Piatetsky-Shapiro G., Frawley W.: *Knowledge Discovery from Databases*. Cambridge MA, 1991.

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 169–175 

# Mining of an electrocardiogram

Urszula Markowska-Kaczmar and Bartosz Kordas

Wroclaw University of Technology, Institute of Applied Informatics, Wroclaw, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

**Abstract.** Widespread use of medical information systems and explosive growth of medical databases require methods for efficient computer assisted analysis. In the paper we focus on the QRS complex detection in electrocardiogram but, the idea of further recognition of anomalies in QRS complexes based on the immunology approach is described, as well. In order to detect QRS complexes the neural network ensemble is proposed. It consists of three neural networks. The details referring to this solution are described. The results of the experimental study are also shown.

## 1 Introduction

An electrocardiogram ($ECG/EKG$) is an electrical recording of the heart activity and is used in the heart diagnoses. ECG allows evaluating the rhythm and frequency of the heart work and enables investigation of the heart defects of people. On the basis of ECG recording one can evaluate the size of the heart chambers. Knowledge of the ECG image for healthy and defect cases is the base for the heart diagnose. The ECG signal provides information concerning electrical phenomena occurring in the heart, which influence on the change of shape of an individual part of the signal. Because of the direct relationship between the ECG waveform and interval of the heart beats, it is possible for the doctor to diagnose cardiac disease and monitor patient conditions from the unusual ECG waveforms. The basic analysis of ECG signal is based on the following elements (Fig. 1): wave P, spike Q, spike R, spike S, wave T. Especially, the complex QRS is essential for every heart anomaly detection. Traditionally, such a detection is done by physicians in order to examine whether the patient heart works properly or not.

In recent years, a significant amount of research effort has been devoted to the automated detection of spikes in ECG signal [1–4]. In general, they can be divided into linear methods of the signal processing and methods of nonlinear transformation. The most important techniques [5] are:

- calculation and analysis of the first derivative of ECG signal,
- pattern matching,
- using DFT or wavelet for discovery of periodic part of the signal [6],[7],
- using digital filter (e. g. FIR, IIR),
- heuristics equations,
- neural networks,
- nonlinear principal component analysis of the ECG signal.

The performance of an automated ECG analysis systems depends heavily on the reliable detection of QRS complex. The difficulties of characteristic waves detection lie in oscillations in the baseline, irregularities of the waveform and frequency overlapping among wide band distribution of the characteristic waves. These arguments explain so many different approaches in an automatic detection of QRS complexes.

In the paper we focus on the automatic detection of the QRS complex using ensemble of neural networks. The QRS complex detection is the starting point for further analysis of ECG signal allowing to determine a kind of anomaly occurring in ECG signal. In other words, QRS complexes (or features acquired from them) are the basis of classification of the ECG signal giving in the result a patient diagnose. Classification is one of the datamining tasks—new emerging technology, which is well suited for the analysis of data. It is becoming an important tool in science, healthcare and medicine.

## 2 The idea of ECG mining

In the presented study, signals from twelve leads of the ECG recording are delivered to the preprocessing module, where they are processed in order to obtain input features for a detection of characteristic points of the signal such as QRS complex, wave P or T.

The signal is represented as the sequence of voltage values relating to the activity of the heart. It is sampled every 4 ms. Because the measurement lasts 5 s. the sequence is composed of 1200 values. The signal is displayed

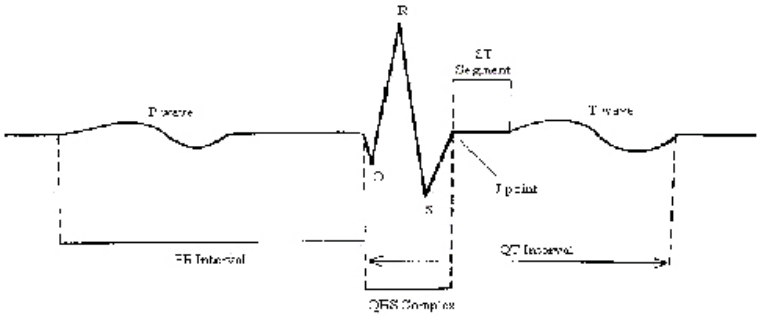



**Fig. 1.** The characteristics of an electrocardiogram

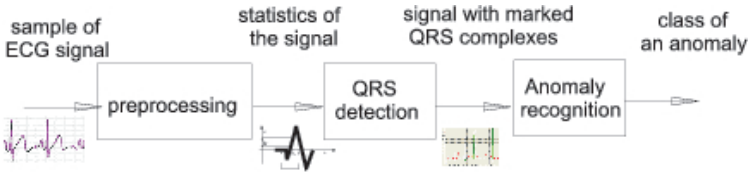



**Fig. 2.** The idea of ECG mining

to the user allowing to mark patterns (parts of the signal) for training the detection module (Fig. 2). This module is based on an ensemble of neural networks. Each neural network in the ensemble has its own input representation of the signal in order to show different aspects of information hidden in the signal. The final decision of the neural network ensemble follows according to the voting procedure, and this decision has to be unanimous.

The output of detection module can be processed further either manually by the physician to diagnose the state of patient or can be delivered to the anomaly recognition module. This module on the basis of the characteristic spikes in the signal (QRS complex, T or P wave) classifies the part of the signal to the proper category. The solution of this problem in our approach is inspired by the immune system. In the paper we concentrate mostly on the detection module, describing its details and presenting obtained results while for the problem of anomaly recognition the idea is sketched only. It determines the direction we will follow in our research in the nearest future.

## 3 Detection of QRS spikes

Due to the differences in the size of the hearts, the orientation of the heart in the body and healthiness of heart itself automatic detection of QRS complex is not an easy task. In order to improve the reliability of the spike detection we used a neural network ensemble. It is composed of three neural networks (NN1, NN2, NN3). Each network is the classical multilayered neural network, trained with backpropagation algorithm with momentum. It classifies actual input pattern into one of the following classes: QRS complex, wave P or T, or meaningless pattern. Therefore there are three neurons in the output layer of each neural network.

The number of input neurons for each network depends on the size of the input pattern. Because we decided to deliver different kind of information about the ECG signal for each neural network, it means that the number of inputs in each network is various according to the size of the input vector. The idea of the neural network ensemble detecting QRS complex is shown in Fig. 3.

It has to be pointed out that the input pattern of each neural network refers to the same part of the signal but it contains other parameters. When the networks are trained, the decision of the detection module about pattern classification is made on the basis of common decision of neural networks in the ensemble. The rule is that decisions of all neural networks have to be unanimous. In other case classification is incorrect and the user is informed about that.

### 3.1 Input data for neural networks

The recorded ECG signal is stored as discrete samples describing an electrical activity of the heart in a given discrete time $t_i$ (Fig. 4). The training set for the neural networks in the ensemble is prepared on the basis of patterns

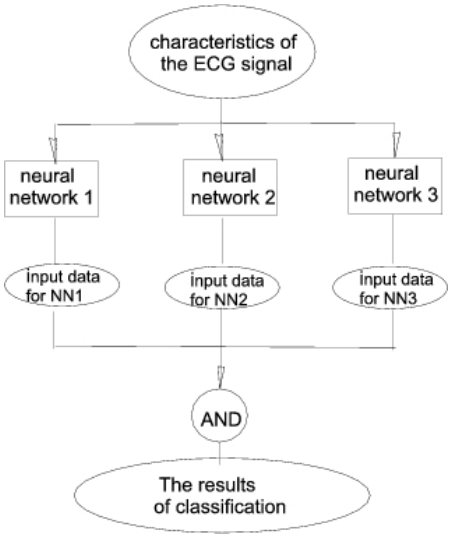



**Fig. 3.** The idea of spike detection in an electrocardiogram

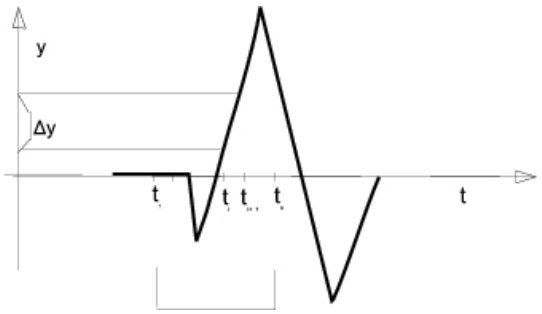



**Fig. 4.** The idea of data preparing for a neural network

in the signal (QRS complex, T and P wave) marked on the screen by an expert. Such a rough representation of the signal is composed of the sequence of real values that are not useful to process by a neural network (NN). That was why we have to preprocess the input data.

**Input data representation for NN1** The aim of the first representation of the signal, which refers to the NN1 is to get hold of similarity between different fragments of signal. The trigonometric representation of the signal was used in this case. It is based on the transformation of the signal from the amplitude values to the sequence of angles (represented by its sinus value). This representation allows to formulate information independently of the absolute signal value while the important characteristic of the signal are kept. It refers to the speed of growth or decline of the signal.

In order to transform the *value(i+1)* of the signal in the timestep *(i+1)* to its angle representation (the value of sinus) it is necessary to calculate the change $\Delta y$ according to the eq. 1:

$$\Delta y = value(i+1) - value(i) \tag{1}$$

Next, the distance between points are calculated as follows:

$$dist = \sqrt{\Delta y^2 + \Delta t^2}. \tag{2}$$

We assumed that each timestep was equal to 1, so the $dist$ can be expressed as:

$$dist = \sqrt{\Delta y^2 + 1}. \tag{3}$$

These two parameters are used in order to formulate the new value expressed as sinus of the angle in the signal:

$$value_{ang}(i) = \Delta y / dist. \tag{4}$$

These could be quite good to detect the QRS complexes in case when they would occur regularly and they would have exactly the same shape. Reality is more complex so we decided to introduce the next networks with other input data.

**Input data representation for NN2** The problem with the first representation occurs when we try to consider the following two patterns as QRS complexes. Let say the first one rises in the range [0,20] and in the range [21,50] it decreases while the second one has its growth in the range [0;30] and the decrease in the range [31;50]. In such case with the first representation neural network would have difficulties to decide what is in the range [21;30]. That was the reason that for the second network the range of angles was discretized into 14 following ranges:

(85, 90); (75, 85); (60, 75); (45, 60); (30, 45); (15, 30); (0, 15);
(-15, 0); (-30, -15); (-45, -30); (-60, -45); (-75, -60); (-85; -75); (-90, -85)

Next, the frequency of the appearance of the given angle range in the sample of signal is calculated. Then the input vector is formed as 14 elements each of which relates to one angle range. The number of ranges = 14 was chosen on basis of experiments. In consequence the number of neurons in the input layer of NN2 is 14, as well.

**Input data representation for NN3** The next input data representation is similar to the previous one. The corresponding ranges of angles are not distinguished and are represented by their absolute value. It means that we have only seven ranges of angles and for example the ranges (85, 90) and (-90, -85), are treated as the same range. In this case for the considered part of signal we obtain the input vector composing of seven elements. Each of them informs about the frequency of the appearance of the given angle range.

While the number of input neurons in networks NN2 and NN3 is fixed, the number of neurons for NN1 is set by the user. Summarizing, the representation of the signal for each network was designed to show some specific features of the ECG signal. The first one helps to discover a similarity between parts of signal and to preserve information about the spike. The second representation delivers information about the function gradient (without knowledge about its location). The last one identifies and distinguishes spikes from flat parts.

### 3.2 Training set for the neural network ensemble

The training set is prepared by the user. He/she marks the appropriate part in the signal displayed on the screen and assigns to it the type of the pattern (QRS spikes or P/T wave or 'meaningless'). Next, the representation of the patterns for each neural network is prepared by the application. It means that for NN1 each pattern has to be scaled according to the number of inputs assumed for this network. For other networks the frequency of ranges is calculated.

### 3.3 Detection of QRS complexes by the neural network ensemble

When the process of the neural network ensemble training ends, the system is ready to discover characteristics of ECG signal (QRS complexes, P and T waves). First, the examined signal has to be loaded. Next, the user decides which type of the signal characteristic should be automatically marked on the screen. The range of size of the moving window has to be defined, as well.

The principle of the system performance is based on the progressive signal processing. The size of the window moving along the $t$ axis starts from the smallest window size and increases during the progress of detection. The step of the window movement and its increase influence on the time of signal processing, but also on an efficiency of the spike detection.

The moving and increasing window enables the detection of patterns independently of its width but the complexity of this algorithm is high. Each of the neural networks gets the transformed sample of the signal from the window and performs its classification. As an output of each neural network, the vector of three elements is produced. Each output is responsible for detecting a suitable characteristic of the signal, ie. QRS complex, wave P or T. In case when the neural networks answers are unanimous, the decision of the neural network ensemble is assigned to the pattern. It is the basis to mark the detected spikes in the image of the signal.

### 3.4 Procedure of detected spike visualisation

After signal processing by the neural network ensemble, we obtain many patterns with an answer assign to each of them (detected QRS complex, wave P/T or meaningless pattern-class of the detected pattern). Because values from

the signal are repeatedly processed in various patterns they can be classified in a different way. In order to mark the spikes sought by the user, the overlapping parts of the signal have to be joined together. A new parameter—*sensitivity* is vital in the process of joining those parts. It is used to decide in how many parts a particular point has to appear to be recognized as a part of marked answer. Next, those points are joined to become a cohesive range—one of the detected intervals. It is a kind of a treshold that allows to decide whether the value belongs to the sought pattern. As we can see in the experimental part of the paper this parameter plays a crucial role in the successful spike detection.

### 3.5 QRS complex detection—experimental studies

In the first phase of experiments we investigated the influence of the neural networks parameters in order to find the values guaranteeing successful results. The number of neurons in the hidden layer was chosen according to the rule of thumb as the average of the number of input and output neurons for each network. The best setting of neural network parameters in the ensemble was as follows: initial weights were randomly chosen in the range $[-0.1; 0.1]$, learning coefficient = 0.001, training error = 0.0001. The last two parameters need some comments. The bigger their values cause that the networks in the ensemble were not unanimous in their decision, so the ensemble was not able to classify patterns properly. For example we observed that for detecting P/T waves it would be better to set learning coefficient even smaller than 0.001, but it extends time of training of neural networks. In further experiments the neural network ensemble was trained using training set composed of 60 training patterns (20 patterns for each class).

The next phase of experiments was focused on the role of the sensitivity parameter. The proper setting of its value is essential for the results produced by the application. It is visualized in Fig. 5. As it can be noticed, the recognition of QRS complex is more accurate when the bigger value of sensitivity is used. The higher is its value, the narrower is the range of detected pattern. In case when the sensitivity value is too small, unwanted regions belong to the marked part of the pattern. The additional difficulty is the necessity of individual tuning of sensitivity value for each ECG signal.

In Fig. 6 the results of QRS recognition are shown for ensemble consisting of two neural networks (a) or three neural networks (b). Comparing both cases, we can conclude that the extended neural network ensemble produces narrower ranges of patterns, what gives more accurate results.

The proper evaluation of the described approach of the QRS complexes detection still needs more experiments, but we can conclude that the application gives satisfying results especially in detection of QRS complexes. The detection of P/T waves was also satisfying, but because their shapes are not so specific comparing to the QRS complexes, sometimes the detection failed.

## 4 Anomaly recognition—future works

The recognized sample of signal (QRS complex) should be properly interpreted in order to acquire additional knowledge (to check whether the person is healthy or not). An expert can easily say whether the signal includes any anomalies. The algorithm of the automatic anomaly recognition described in this document requires samples of ECG signal, but can cope with recognizing anomaly using only positive patterns. It is inspired by a natural immune system activity based on negative selection phenomenon (self-nonself discrimination process that takes place in thymus), which is also an origin of Negative Selection Algorithm (NSA), discussed wider in [8], [2] and [9].

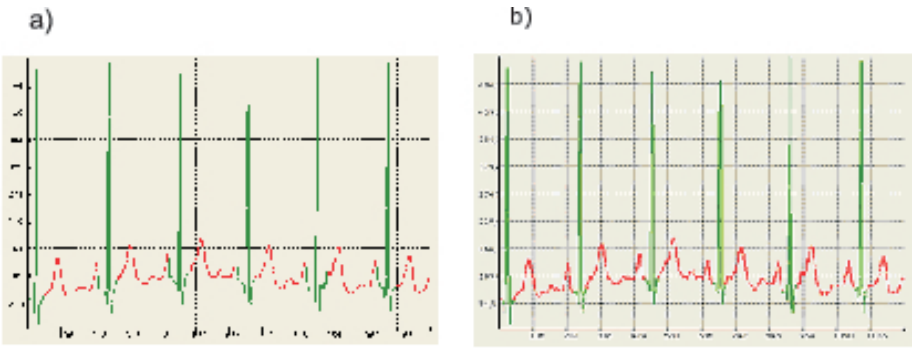



**Fig. 5.** The comparison of QRS complex recognition for two cases a) sensitivity=10 and b) sensitivity=100

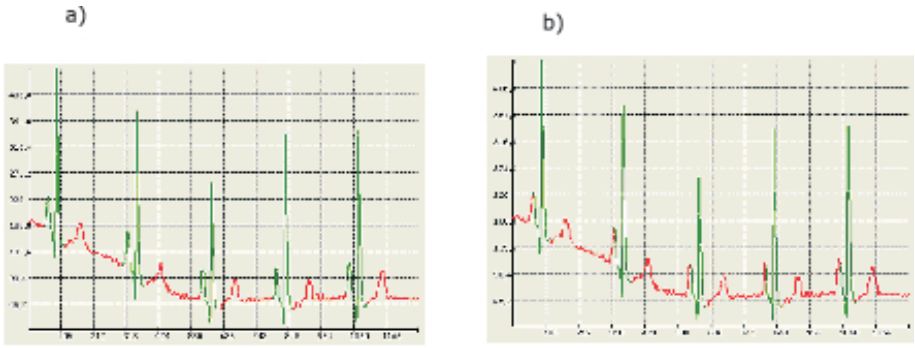



**Fig. 6.** The results of the neural network ensemble a) composing of two neural networks and b) three neural networks

Parts of the signal corresponding to QRS complexes have to be recognized by immune system receptor, in order to determine its membership in the group of proper or abnormal structures. Both—samples of signals and receptors consist of strings (binary or real values—depending on the problem idiosyncracy). The initial set of receptors is created basing on model signals transformed by random mutations (fluctuations in higher or lower level). Next, those receptors from the previously created set, that have ability to properly recognize anomalous parts of signal, are selected.

The process of choosing those effective receptors rests on stimulating them with self structures (proper signals samples). Those receptors, which do not recognize any of the self structures are added to the effective receptors set (see Fig. 7) and shall become the base of anomaly recognition algorithm performance. Anomaly recognition relies on the presenting already recognized QRS complex to the NSA system. In the case of positive answer (recognition by any receptor), the signal presented to the receptor is marked as abnormal. That fact is alerted by the system. The key of the NSA is affinity function, which is a measure of receptor and presented structure similarity.

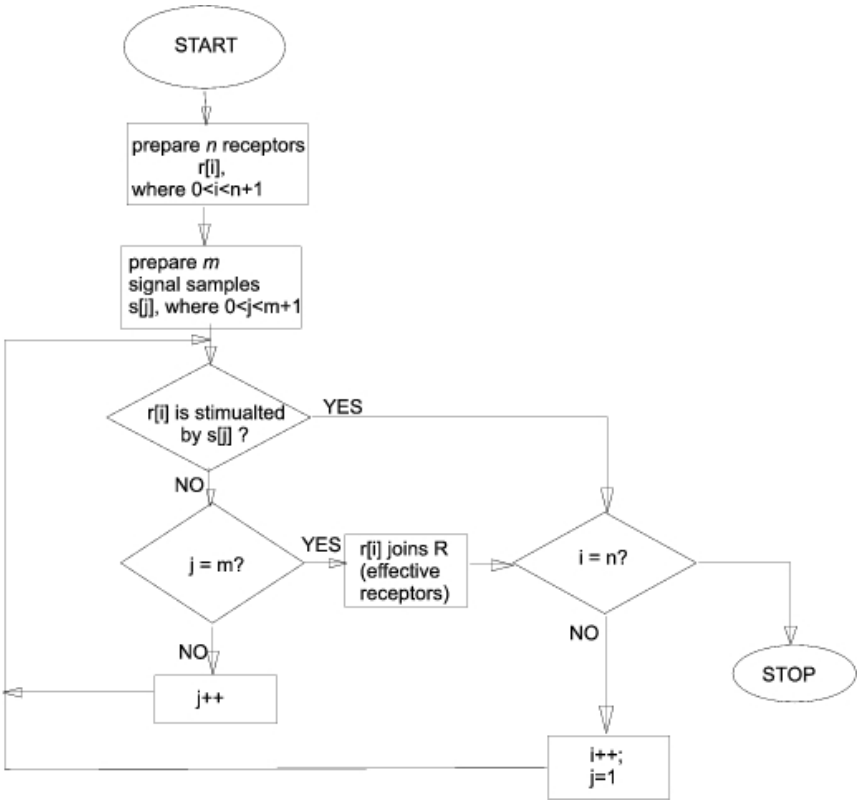



**Fig. 7.** The algorithm of effective receptors set creation

Lack of model signals may become a problem. A solution presented below tries to avoid necessity of possessing many model signals and allows the algorithm to work using just one positive sample. In that case anomaly detection becomes more complex, because creation of both positive and negative samples is necessary. The idea of NSA modification, presented in the paper reaches out the problem. The modification idea rests on using a set of positive detectors, which work just opposite than NSA receptors. The first, original detector is an exact copy of model sample signal for QRS complex of a lead. Next, using random mutations, new patterns are created. Those new patterns are presented to the detectors. In the case of detector's stimulation by the pattern, it is marked as self structure, otherwise nonself structure. Self structures join detectors set. The rejected ones however join the nonself structures set. This process lasts as long, as the cardinality of detectors set does not reach previously assumed value. The algorithm can be compared to negative selection phenomenon, which takes place in thymus, with one main difference: detectors work opposite than receptors (the idea is inspired by T cells performance, whose job is to detect nonself structures) and its set is widen by those detectors, the structure of which is similar to one (or more) of other detectors.

In order to find out whether a sample signal detected (by an expert or neural networks ensemble) as QRS complex includes an anomaly, it is introduced to the described immune system (with already prepared set of detectors and nonself structures) and the system assigns it to one of the groups - detectors (self) or nonself.

## 5 Conclusions

In the paper the idea of ECG mining is presented. It consists of the QRS complex detection and anomaly recognition in samples of the ECG signal. To detect QRS complex we propose neural network ensemble. Each network in the ensemble obtains information about the signal expressed in various form. The results obtained during experiments with the application were promising, but to evaluate the proposed approach in all extend further experiments are necessary. All details referring to the proposed QRS detection are precisely described in the paper. The idea of anomaly recognition based on immune system is proposed, as well, but its implementation and experiments are planned in the nearest future.

## References

1. Fernandez J., Harrisand M., C. Meyer: *Combing algorithms in automatic detection of r-peaks in ecg signals,* In: 18 th IEEE Symposium on Computer-Based Medical Systems (CBMS'05). (2005) pp. 297–302.
2. Gonzalez F., Dasgupta D., Kozma R.: *Combining negative selection and classification techniques for anomaly detection,* Journal of Genetic Programming and Evolvable Machines **4** (2003) pp. 383–403.
3. Kohler N. U., Hennig C., Orglmeister R.: *The principle of software qrs detection,* IEEE Engineering in Medicine and Biology Magazine **21** (2002) pp. 42–57.
4. Thakor N., Webster J., Tomkins W.: *Optimal qrs detector,* Medical and Biological Engineering and Computing **21** (1993) 343–359.
5. Friesen G. M., Jannett T. C., Jadallah M. A., Quint S. R., Nagle H. T.: *A comparison of the noise sensitivity of nine qrs detection algorithms,* IEEE Transaction on Biomedical Engineering **37** (1990) 85–98.
6. Froese T.: *Classifying ecg data using discrete wavelet transforms,* In: Symposium for Cybernetics Annual Research Projects SCARP'04. (2004).
7. Li C., Zheng C., Tai C.: *Detection of ecg characteristic points using wavelet transforms,* IEEE Transactions on Biomedical Engineering **42** (1995) 21–28.
8. Forrest S., Perelson A. S., Allen L., Cherukiri R.: *Self-nonself discrimination in a computer,* Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy (1994) 202–212.
9. Wierzchon S.: *Sztuczne Systemy Immunologiczne. Teoria i zastosowania,* Akademicka Oficyna Wydawnicza Exit, Warszawa (2001).

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 177–185 

# Rule Extraction from Neural Network by Hierarchical Multiobjective Genetic Algorithm

Urszula Markowska-Kaczmar and Krystyna Mularczyk

Wroclaw University of Technology, Institute of Applied Informatics, Wroclaw, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

**Abstract.** The paper presents a method of rule extraction from a trained neural network by means of a genetic algorithm. The multiobjective approach is used to suit the nature of the problem, since different criteria (accuracy, complexity) may be taken into account during the search for a satisfying solution. The use of a hierarchical algorithm aims at reducing the complexity of the problem and thus enhancing the method's performance. The overall structure and details of the algorithm as well as the results of experiments performed on popular benchmark data sets are presented.

## 1 Introduction

The problem of extracting rules from neural networks consists in finding the dependency between the network's output and the properties of the presented input vector [1]. It is very important in medical domain because it increases the trust of users to the system which uses neural network. The complexity of this task arises usually from a large number of dimensions of the input vectors. The usefulness of genetic algorithms results mainly from their ability of searching vast multidimensional spaces, as well as processing many potential solutions at a time. Moreover, unlike other methods that treat the network as a "black box" and concentrate only on the values of inputs and corresponding outputs, thus becoming architecture independent (e. g. decision trees), genetic algorithms prove to be more successful when dealing with real values of inputs. Several techniques, usually for networks whose task is to classify the input vectors, have already been developed, e. g. in [2, 3] however, the multiobjective nature of the problem of rule extraction is rarely taken into account. A set of rules describing the functioning of a network may be optimised in several various ways. First of all, its accuracy may be improved by modifications that lead to maximizing the number of correctly classified patterns as well as reducing the number of errors made by the set. Another criterion taken usually into consideration is the complexity of the set, denoted by the number of rules and the number of premises in each rule [1], which in turn may be considered separately.
This implies the existence of several criteria of the evaluation of the solutions that tend to be contradictory. The standard approach in a genetic algorithm is based on the concept of aggregating functions that enable gathering all the values of the criteria into a single fitness function, usually in the form of a weighted sum. However, this method has two main drawbacks. The weights must be properly adjusted to allow the algorithm to find satisfactory solutions, which may result in the necessity of running the algorithm many times. Moreover, whenever the user decides to change the weights in order to concentrate on one criterion to a greater degree, the process of evolution must be repeated. This makes the use of classic genetic algorithms rather unwieldy and time-consuming.
During the process of rule extraction one may wish to take one of different approaches, depending on the purpose of extraction. One option is an attempt to obtain a relatively simple set of rules that contains only the most significant principles governing the functioning of the network, without focusing on exceptions. Another approach aims at developing a set that describes the network with the highest fidelity possible. This might lead to discovering "hidden" knowledge by finding rules that apply to small numbers of input vectors. Taking this into account, one may say that a perfect solution to the problem of rule extraction would produce results at a different level of accuracy and complexity to suit all the needs. This requirement is met by multiobjective genetic algorithms. Such a method has been proposed in [4] but, since the algorithm operates on entire sets of rules, its efficiency when dealing with complex problems is questionable.

## 2 Pareto optimality in multiobjective problems

In a multiobjective problem several criteria are used to evaluate each solution. The quality of a given solution cannot be easily expressed in terms of a single numeric value; hence the problem of comparing solutions and determining which of them proves to be the most satisfactory. The concept of Pareto optimality introduces a relation of quasi order on the set of solutions, called Pareto domination and denoted by the symbol $\prec$. For two solutions $x$ and $y$, whose quality is measured by two vectors consisting of the values of individual criteria, namely:

$$f(x) = [f_1(x), f_2(x), \ldots, f_m(x)], \tag{1}$$

$$f(y) = [f_1(y), f_2(y), \ldots, f_m(y)], \tag{2}$$

the relation of domination is defined as follows:

$$f(x) \prec f(y) \Leftrightarrow (\forall k = 1, \ldots, m)\ f_k(x) \leq f_k(y) \\ \wedge\ \exists\ f_k(x)) < f_k(y). \tag{3}$$

It means that a given solution dominates another if it is better with regard to one criterion and at least equally good if any of the remaining criteria is taken into account. This implies that in a general case two solutions do not necessarily have to be bound by this relation, which makes them equally valuable. The optimisation in the Pareto sense, unlike the optimisation of a function, does not aim at producing a single satisfactory solution, but at finding an entire set of non-dominated solutions that would not only lie possibly close to the optimal values, but would also be distributed evenly in the space of solutions. The second condition guarantees that all criteria and various combinations of their relative "importance" are equally taken into account during the process of optimisation.

## 3 Basic concepts of the proposed method

The method of rule extraction proposed in the paper, we called MulGEx (**Mul**tiobjectve **G**enetic **Ex**tractor) is based on a genetic approach. It acquires a set of rules describing the performance of a network is used for classification tasks. Although we concentrate on rule extraction from a neural network, since MulGEx treats a network as a black box (a global approach to rule extraction [1]), it can be used for extracting sets of rules immediately from raw data as well. The input vector presented to the network may consist of data of various types (logical, enumerable, real) whereas the only output is an integer representing the class that a given pattern belongs to. Because of the great complexity of the problem, especially for large numbers of attributes and input vectors, the algorithm has been divided into two stages, which had been inspired by [2].

The lower-level genetic algorithm operates on single rules and aims at maximizing the number of correctly classified patterns without taking the rules' complexity into account. The upper-level algorithm, on the contrary, respects all criteria mentioned in the introduction. Its task is to gather the rules evolved by the first algorithm into one set and refine them through further processing. The general structure of the algorithm has been shown in Fig. 1. The chromosome on the lower level represents a single rule consisting of a set of premises followed by the conclusion, as shown in Fig. 3a. A premise may be active or not, depending on the value of its flag. More detail and an explanation of the purpose of flags is given in the next subsection. Similarly, a set evolved by the upper-level algorithm is a sequence of rules that may also be activated or disabled when the corresponding flag is changed. The chromosome at this stage represents a set of rules. Its structure is presented in Fig. 3b.

### 3.1 The description of the lower-level algorithm

The lower-level algorithm consists of several independently evolved populations, each responsible for a different class in the problem of classification. At this stage an individual represents a single rule in the form of a set of premises followed by the number of the appropriate class, as shown in 4.

$$IF\ p_1\ AND\ p_2\ ,\ldots\ AND\ p_m\ THEN\ k \tag{4}$$

Each premise $p_i$ corresponds to one element of an input vector presented to the network (the $i$-th input of the network)and denotes the constraint that the attribute must satisfy before the rule may be applied to the entire vector. Depending on the type of the attribute, this constraint may take one of the following forms.

- *For logical attributes* the constraint determines which of the two possible values must be taken by the attribute. Therefore the premise consists of a single Boolean value (Fig. 2a).
- *For enumerable attributes* the premise is expressed as an array of Boolean values ($V_i$) with one element per every value in the type. It determines the subset of values accepted by the constraint (Fig. 2b)
- *Discrete and real attributes* require defining the range of acceptable values, which is achieved by specifying its minimum and maximum
- *Discrete and real attributes* require defining the range of acceptable values, which is achieved by specifying its minimum and maximum (Fig. 2c).

Every rule contains a complete set of premises, i. e. imposes constraints on all elements of input vectors presented to the network. This prevents from introducing chromosomes with variable lengths and thus facilitates defining the genetic operators. However, in many cases certain attributes do not influence the choice of a class

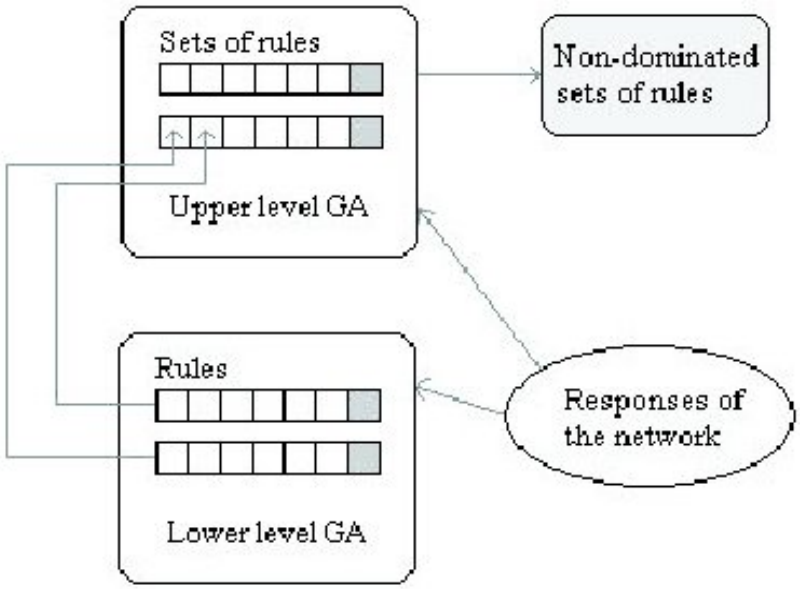



**Fig. 1.** The general structure of MulGEx

during the process of classification, and therefore all their values could be accepted by a given rule. To achieve this, a Boolean flag is added to every premise. It determines whether the particular constraint remains active and is taken into account when the rule is applied to an input vector. By disabling some of the constraints one may obtain more general rules with lesser complexity, which is essential during the process of rule extraction.

The form of the chromosome (rule) requires defining special genetic operators responsible for crossover and mutation. Uniform crossover has been used, i. e. the process of exchanging genetic information between two individuals consists in choosing one of the parents with equal probability for each premise (gene) copied to the chromosome of the offspring. A premise is duplicated entirely, including the flag. The only exception occurs in the case of constraints imposed on real inputs, where each of the values determining the accepted range (namely the minimum and maximum) may be taken from a different parent. The conclusion of the rule may be copied from either of the parents, since the number of the class is identical for all individuals in a given population.

The effect of mutation depends on the type of the gene that it's applied to. In all cases the flag may be affected, which results in enabling or disabling a given gene. For the types consisting of Boolean values shown in Fig. 2a and Fig. 2b, mutation may cause changing a randomly chosen value to the opposite one. The operator of mutation applied to a real value in a premise of the type shown in Fig. 2c changes it by a random amount (without exceeding the maximal range defined for the corresponding attribute), on the assumption that small modifications are more probable. Mutation doesn't affect the conclusion of a rule.

In order to compare both approaches (scalar and multiobjective), the value of the fitness function assigned to an individual in the lower-level algorithm may be either calculated as a weighed sum of the criteria or based on the relation of domination (the multiobjective approach). The first step in both cases consists in determining

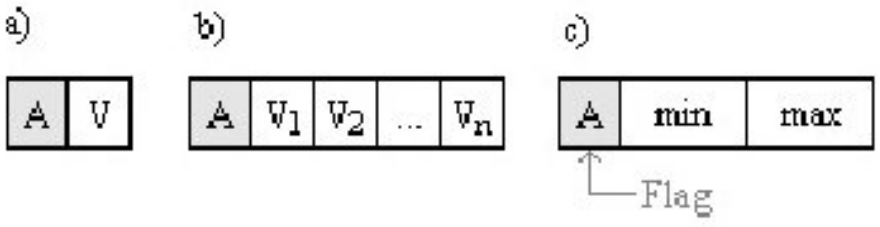



**Fig. 2.** Genes for different type of premises, which depending on the type of attribute, a)—for logical, b)—for enumerative, c)—for real attribute.

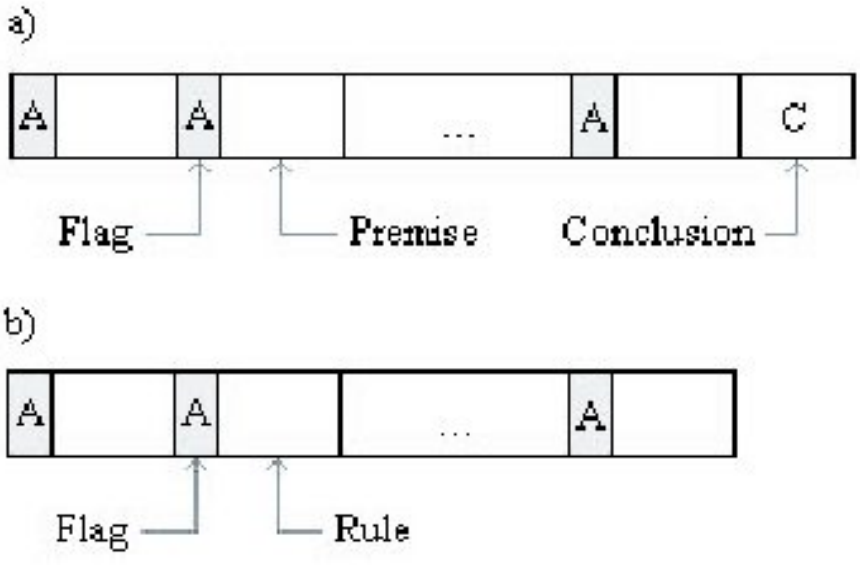



**Fig. 3.** A schema of chromosomes on both levels

the number of correctly and incorrectly classified input vectors, i.e. the *coverage* and *error*, respectively. To this end the answer given by the neural network for each vector is compared to the answer given by the rule if the constraints have been satisfied. The weights used for fitness calculation in the first approach are chosen by the user and the function takes the form presented by Eq. 5.

$$Fitness_{lower} = W_c \cdot coverage - W_e \cdot error \tag{5}$$

If the value of the function happens to be negative, it must be modified, e. g. by adding a constant value to the fitness of every individual.

In the multiobjective approach, the fitness assignment is based on ranks given to the individuals. Several techniques may be used here—the most popular were proposed by Goldberg in [5] and Fonseca and Fleming in [6]. Both of them are implemented in MulGEx, the latter with a modification introduced by Zitzler and Thiele (the SPEA algorithm [7]). The selection of individuals that participate in the process of reproduction is performed by means of the roulette wheel method, which implies that the probability of choosing a given individual for crossover is proportional to its fitness. Goldberg's algorithm of rank assignment is as follows:

$Temp := Population$
$n = 0$
$while(Temp \neq 0)\ do$
{
$Nondominated = \{x \in Temp\ /\ (\neg \exists\ \mathrm{y} \in Temp) f(y) \prec f(x)\}$
$(\forall x \in Nondominated)\ \mathrm{rank}(x) := n$
$Temp := Temp \setminus Nondominated$
$n := n + 1$
}

Fitness assignment in SPEA is performed according to the following algorithm:

*Let N denote the size of the population*

*For each individual* i *in the external set* :
$f_i = \frac{N}{N+1}$ ,
*where n stands for the number of individuals*
*dominated by i* $f_i \in [0, 1)$

$For\ each\ individual$ j $in\ the\ population$ :
$f_j = \sum_{i,i \prec j}\ f_i + 1$,
$i.e.\ the\ sum\ of\ fitness\ values\ of\ all\ external$
$individuals\ that\ dominate\ j\ increased\ by\ one$
$f_j\ \in [1, N)$

Because in this method the lowest fitness values are assigned to the best individuals, the fitness function must be modified so that it can be used for the roulette wheel technique.

Since the complexity criterion is not taken into account during the fitness assignment at this stage, all genes of the individuals in the initial population are set to disabled. During the process of evolution the premises are gradually activated by applying mutation, which allows to keep the complexity at a relatively low level and introduce only the necessary constraints. The duration of evolution is determined by the user who may choose either to create a fixed number of generations or to stop the algorithm when no improvement has been detected for a given amount of time (measured in generations). When the algorithm has stopped, the best rule in every population is stored in an external set and all vectors recognized by this rule are removed from the set that the algorithm works on. A good idea is to choose the rule with the highest coverage and no error, but in some cases this might be impossible or inefficient. Afterwards a new initial population is created and all steps are repeated. This phase of rule extraction ends when there are no more vectors corresponding to a given class in the set, so that no new rules may be evolved. The external set containing rules stored during the entire course of evolution is passed to the upper-level algorithm for further processing and optimisation.

### 3.2 The description of the upper-level algorithm

The upper-level algorithm is used to refine and optimise the rules evolved by the previous algorithm. This is achieved by gathering them into a single set and evaluating the efficiency of their cooperation. As a result, superfluous rules may be eliminated and the remaining ones simplified. Because at this stage a multiobjective algorithm is used, the evolved sets of rules have different features; they vary both in accuracy and complexity. The most general structure of the algorithm resembles the one introduced at the lower level where, an individual represented a rule consisting of premises that could remain active or inactive. In this case a chromosome codes a set of rules that may be disabled in the same way, which excludes them temporarily from participating in classification. Again, in order to avoid operating on chromosomes with variable length, each rule is accompanied by a flag determining its status in the set. In the initial population all individuals are exact copies of the set of rules obtained from the lower-level algorithm. This implies that the maximal accuracy is available without the necessity for adding new rules. The only condition that has to be met is that all genes remain active. Therefore the task of the algorithm is limited to searching for sets with reduced complexity. However, the improvement of this criterion should be accompanied by the lowest possible deterioration of accuracy.

The process of reproduction, just like at the earlier stage, consists in exchanging genes in a random way between the parents (uniform crossover). This means that every rule and its flag in a newly created individual are copied from the parent that has been selected randomly for this purpose. Mutation occurs on two levels. First of all, a randomly selected flag may be changed, which results in altering the activation status of the corresponding rule. The effect is that a potentially superfluous rule is excluded from the set or, in the opposite case, a previously removed one is restored. The other type of mutation affects single premises inside the rules and is performed in the same way as in the lower-level algorithm, thus leading to the modification of constraints imposed on the attributes. Since in the previous algorithm some rules were evolved on the basis of a reduced set of input vectors, at this stage they have a chance to improve their quality through the use of the entire vector set as a ground for evaluation, as well as cooperation with other rules.

The upper-level algorithm is multiobjective and therefore its fitness function is based on the relation of domination. The entire process of selection, including the rank-based method of assigning fitness to an individual as well as the technique of the roulette wheel, is identical to the one used in the lower-level algorithm. The only difference consists in the choice of criteria for the evaluation of individuals and the way of calculating their values. The coverage of a set is determined by the number of input patterns that are correctly classified by the set as a whole, i.e. by at least one of its active rules. The error, on the contrary, depends not only on the number of misclassified vectors, but also on the frequency of a given mistake. In other words, the error is calculated as the sum of errors for all active rules in the set. On the upper-level another criterion, namely the complexity of a set, is added. Its value is determined by the number of active rules in a set increased by the number of active premises in each of them. This criterion could be split into two, but, apart from complicating the algorithm, such a distinction would not result in a significant improvement of performance.

The process of evolution stops after evolving a given number of generations measured either from the beginning or since the last improvement observed (like in the previous algorithm). However, the definition of improvement in a multiobjective algorithm is not as straightforward as in classic algorithms, where the average fitness of the population may be easily compared in the course of evolution. In this case the quality of an individual is expressed as a vector consisting of several values. Average values for all criteria in the population may be calculated, but the comparison of different generations must be based on domination. In other words, a new generation is recognized as more fit only if the vector of those average values dominates a corresponding vector calculated for the previous generation. The output produced by the algorithm is a set of non-dominated solutions. The choice of the most appropriate one (in a given situation) is left to the user, who may apply certain weights to the criteria and obtain the solution with the highest weighed sum of their values. This might be done repeatedly, which results in presenting solutions with various features without the need for rerunning the algorithm with different parameters.

## 4 Experiments

The purpose of performing experiments is to verify whether the proposed method can be applied to various problems and whether it remains efficient regardless of the number and types of attributes in a set of input vectors. The use of popular benchmark data sets gathered in [8] during the tests facilitates comparing the method with other techniques developed for a similar class of problems. Table 1 presents detailed information about the data sets that have been used for the tests, namely: $Iris, LED-24, Monk-1$ and $Quadrupeds$. A 2 % noise has been introduced to the $LED-24$ data set.

**Table 1.** Data sets used for the tests

| **Name (examples)** | **Type of attributes** | **Class** | **No of examples** |
|---|---|---|---|
| Iris (150 ) | 3 continuous | Setosa | 50 |
| | | Versicolour | 50 |
| | | Virginica | 50 |
| LED-24 (1000) | 24 binary 2% noise | 10 classes | $\approx$ 100/ class |
| Monk-1 (432 ) | 6 enumerable | 0 | 216 |
| | | 1 | 216 |

Separate experiments were carried out directly on the data sets obtained from the repository [8], without having them processed by a neural network beforehand. One more data set (Quadrupeds) was added to verify the

**Table 2.** Results of the network training

| **data (set)** | **Number of neurons in the hidden layer** | **Accuracy %** |
|---|---|---|
| Iris | 3 | 98 |
| LED-24 | 5 | 95 |
| Monk-1 | 3 | 100 |

scalability of the presented method. The reason is that a network tends to simplify data by eliminating noise at least partially. This should theoretically lead to reducing the complexity of rule sets and improving the efficiency of the algorithm, which has been observed during the tests. Extracting rules from data processed by a network resulted in improved efficiency of the method and enhanced accuracy of the solutions.

The proposed genetic algorithm was tested on data that had been processed by a multi-layer feed-forward neural network with one hidden layer, trained by the back-propagation algorithm. The accuracy achieved by the network in each case is presented in Table 2.

During the tests the lower-level algorithm worked with best effectiveness with a relatively low probability of crossover, not exceeding 50%, and a low probability of mutation, up to 1%. The desirable size of a single population depended strongly on the complexity of the problem, particularly on the number of attributes, and was

**Table 3.** Rules for $Iris$ evolved by the lower-level algorithm; *) indicates that coverage has been calculated on the basis of a reduced set of input patterns)

| No | Rule | Coverage |
|---|---|---|
| 1. | **IF** PetalLength ∈ [1.00; 2.90] **THEN** Setosa | 50 |
| 2. | **IF** PetalLength ∈ [1.97; 4.98] **AND** PetalWidth ∈ [0.96; 1.65] **THEN** Versicolour | 47 |
| 3. | **IF** SepalLength ∈ [6.19; 7.17] **AND** SepalWidth ∈ [2.66; 3.32] **AND** PetalLength ∈ [1.00; 5.51]**AND** PetalWidth∈ [0.88; 1.71] **THEN** Versicolour | 2* |
| 4. | **IF** PetalWidth ∈ [1.72; 2.50] **THEN** Virginica | 46 |
| 5. | **IF** SepalLength ∈ [4.30; 6.12] **AND** SepalWidth ∈ [2.00;2.75] **AND** PetalLength ∈ [4.37; 5.76]**AND** PetalWidth∈ [1.37; 2.50] **THEN** Virginica | 2* |
| 6. | **IF** PetalLength ∈ [5.33; 6.27] **THEN** Virginica | 1* |

**Table 4.** The two examples of sets of rules for $Iris$ evolved by the upper-level algorithm (*cov*—stands for coverage, *compl*—stands for complexity)

| No | Set contents | Cov. | Error | Compl. |
|---|---|---|---|---|
| 1. | **IF** PetalLength ∈ [1.00; 2.90] **THEN** Setosa<br>**IF** PetalLength ∈ [1.97; 4.98] **AND** PetalWidth ∈ [0.96; 1.65] **THEN** Versicolour<br>**IF** PetalWidth ∈ [1.72; 2.50] **THEN** Virginica | 143 | 0 | 7 |
| 2. | **IF** PetalLength ∈ [1.00; 2.90] **THEN** Setosa<br>**IF** PetalLength ∈ [1.97; 4.98] **AND** PetalWidth ∈ [0.96; 1.65] **THEN** Versicolour<br>**IF** SepalLength ∈ [6.19; 7.17] **AND** PetalWidth ∈ [0.88; 1.71] **THEN** Versicolour<br>**IF** PetalWidth ∈ [1.72; 2.50] **THEN** Virginica<br>**IF** SepalWidth ∈ [2.00; 2.62] **AND** PetalWidth ∈ [1.37; 2.50] **THEN** Virginica | 148 | 2 | 13 |

usually set to 50 or 100 individuals. Increasing the size of a population (within reason) resulted in better efficiency of the search. On the upper level the probability of crossover and mutation was usually the same. Otherwise the changes in the chromosome became too rapid for the algorithm to be able to refine the rules through small modifications. At both stages of the algorithm its efficiency could be enhanced by adding the option of saving the best individuals in the population. In a multiobjective algorithm this is achieved by introducing an external population consisting of all non-dominated individuals found so far during the course of evolution, as described in [7] (SPEA algorithm). This permits to increase the probability of mutation without the risk of destroying the best individuals. In all cases the process of evolution was stopped when no improvement had been observed in the last 100 generations.

An example of rules for $Iris$ evolved by the lower-stage algorithm for data processed by the network is shown in Table 3. In this case the first rule in each class usually covers the majority of input patterns, whereas the remaining rules supplement it by covering all omitted examples. It's worth noticing that the number of patterns covered by each rule is calculated on the basis of a reduced set, after all vectors recognized by the previous rule have been excluded. Therefore the actual number of examples covered by some of the rules may be larger than indicated. In all cases the number of misclassified examples is equal to zero, which results from the method of choosing the best rule (a very high weight assigned to error). The number of decimal places within the premises has been reduced for clarity. Table 4 presents examples of sets obtained from the upper-level algorithm on the basis of the previously evolved rules. It is apparent that the improvement in accuracy is accompanied by increasing complexity of sets and rules—and vice versa.

The general features of rule sets obtained by the method in question for the chosen benchmark data sets are gathered in Table 5 and Table 6—for data passed through a network and for raw data, respectively. The algorithm has been run five times for every data set and the average values of accuracy corresponding to particular sizes of rule sets have been calculated. In each case the algorithm evolved a wide range of sets with various features, including complex sets classifying correctly all the instances, as well as relatively simple solutions with few rules for each category. This proves that the proposed method is suitable for different types of attributes. Moreover,

very good results obtained for Quadrupeds are the evidence for the method's scalability, i.e. the ability of solving problems with large numbers of attributes.

The results for Iris, LED-24 and Quadrupeds indicate that a set consisting of one rule per category may cover most of the instances although some level of misclassification is possible. Additional rules improve the accuracy in a relatively low degree. This applies especially to the LED-24 data set, where the accuracy of 96% and 92% was achieved by a set of 10 rules whereas a set classifying correctly all input vectors contained 23 rules for data processed by the network and over 60 rules for raw data (not presented in the table). The observed disproportion is caused by noise that requires very specific rules to deal with certain examples. A small increase in accuracy observed when new rules are added to a set indicates that the algorithm starts taking exceptions into account, which is usually not desirable—unless these specific rules lead to discovering "hidden" knowledge, i. e. unknown dependencies concerning small subsets of instances. Therefore the most complex rule sets evolved for Iris, LED-24 and Quadrupeds have little value. This observation is possible due to the multiobjective approach used in the algorithm that allows to develop different solutions in parallel, thus facilitating their analysis and comparison. The results produced by the presented algorithm may be confronted with those obtained for a different multiobjective algorithm (GenPar) described in [2]. Since the same data sets have been used in both cases, it is possible to compare the effectiveness of the methods. However, this can be done to a certain degree only because of differences in the way of performing experiments. The most significant one is that the results presented in [4] and copied to Table 6 contain information on the number of correctly classified instances only (fidelity), without providing any data about the error made by a given set. Moreover, no superfluous attributes had been added to the $LED$ data set.

The results for GenPar, shown in Table 7 prove to be less accurate than those for MulGEx. It is worth noticing that GenPar did not produce a set with maximal possible accuracy in any of these cases. This might be caused by the fact that it operates on entire sets only, which hinders individual rules from being intensively optimised (the lower-level algorithm in the proposed method is aimed solely at improving single rules), as well as by combining coverage and error into one criterion. The presented algorithm, on the contrary, considers them separately, thus gaining the ability of finding a solution with maximal coverage which is followed by modifications leading to the reduction of error.

## 5 Conclusions

The results of the performed experiments indicate that the proposed method shows good effectiveness when extracting rules from various data, with different numbers and types of attributes. It enables producing a wide range of sets of rules, varying from ones with perfect accuracy achieved at the cost of high complexity to relatively simple ones that cover only the most typical cases, as well as dealing with the problem of noisy data. This is obtained by combining a hierarchical genetic algorithm with a multiobjective approach based on the concept of Pareto optimality. The method is architecture independent. It can be used for any classification problem performed by a neural network regardless of its structure, as long as the responses given by the network for the examined set of input vectors are known. The main weakness of this method consists in the necessity of adjusting parameters for both genetic algorithms in order to ensure the effectiveness of evolution. The optimal values of parameters may be different for each problem and no technique of their precise estimation has been developed. Another disadvantage is that the algorithm needs to examine the entire set of input vectors repeatedly to calculate the values of the criteria for each individual. Even if an effective data representation is implemented, large sets cause the deterioration of efficiency. In spite of these drawbacks that are present in many GA-based methods of rule extraction, the results produced by the proposed algorithm may be considered satisfactory.

**Table 5.** The results of experiments ($size$—number of rules, $acc$—accuracy as the difference between coverage and error) performed on data processed by a neural network

| Iris | | LED-24 | | Monk-1 | |
|---|---|---|---|---|---|
| $size$ | $acc$ | $size$ | $acc$ | $size$ | $acc$ |
| 1 | 50 | 1 | 106 | 1 | 72 |
| 2 | 97 | 5 | 507 | 2 | 144 |
| 3 | 141 | 10 | 926 | 3 | 216 |
| 5 | 147 | 15 | 943 | 5 | 360 |
| 8 | 150 | 20 | 953 | 7 | 432 |

**Table 6.** The results of experiments ($size$—number of rules, $acc$—accuracy as the difference between coverage and error) performed on raw data

| Iris | | LED-24 | | Monk-1 | | quadrupeds | |
|---|---|---|---|---|---|---|---|
| $size$ | $acc$ | $size$ | $acc$ | $size$ | $acc$ | $size$ | $acc$ |
| 1 | 50 | 1 | 106 | 1 | 72 | 1 | 144 |
| 2 | 97 | 5 | 507 | 2 | 144 | 2 | 277 |
| 3 | 141 | 10 | 926 | 3 | 216 | 3 | 388 |
| 5 | 147 | 15 | 943 | 5 | 360 | 4 | 497 |
| 8 | 150 | 20 | 953 | 7 | 432 | 6 | 500 |

**Table 7.** The results of experiments for GenPar ($size$—number of rules, $fid$—fidelity equivalent to coverage) on the basis of [4]

| Iris | | LED-24 | | Quadrupeds | |
|---|---|---|---|---|---|
| $size$ | $fid$ | $size$ | $fid$ | $size$ | $fid$ |
| 1 | 50 | 1 | 87 | 2 | 182 |
| 2 | 94 | 4 | 324 | 4 | 471 |
| 4 | 137 | 5 | 473 | — | – |
| 5 | 144 | 13 | 928 | – | – |

## References


1. Andrews R.: *Survey and critique of techniques for extracting rules from trained artificial neural networks,* Knowledge-Based Systems **8** (1995) pp. 1–100.
2. Markowska-Kaczmar U., Zagorski R.: *Hierarchical evolutionary algorithm in the rule extraction from neural network,* ACM Computing SurveysInternational Symposium on Engineering of Intelligent Systems (2004).
3. Santos R., Nievola J., Freitas A.: *Extracting comprehensible rules from neural networks via genetic algorithms,* Symp. on Combinations of Evolutionary Computation and Neural Network **1** (2000) pp. 130–139.
4. Markowska-Kaczmar U., Wnuk-Lipinski P.: *Rule extraction from neural network by genetic algorithm with pareto optimization,* In Rutkowski L., ed.: Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence (2004) 450–455.
5. Goldberg D. E.: *Genetic Algorithms in Search, Optimization and machine* Addison-Wesley Publishing Company, Reading, Massachusetts (1989).
6. Fonseca C. M., Fleming P. J.: *Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization,* In S. Forrest Ed., Proceedings of the Fifth International Conference on Genetic Algorithms (1993) pp. 416–423.
7. Zitzler E., Thiele L.: Multiobjective evolutionary algorithms: A comparative case study and strength pareto approach. IEEE Transaction on Evolutionary Computation **3** (1999) pp. 257–271.
8. Blake C. C., Merz C.: *Uci repository of machine learning databases,* University of California, Irvine, Dept. of Information and Computer Sciences (1998).

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 187–194 

# Formalna weryfikacja w procesie projektowania systemów informatycznych. Projekt COSMA

Jerzy Mieścicki

Instytut Informatyki, Politechnika Warszawska
00 665 Warszawa, ul. Nowowiejska 15/19
email: J.Miescicki@ii.pw.edu.pl

**Abstract.** W artykule omawia się korzyści i trudności, jakie wiążą się z próbami przerzucenia pomostu między praktyką projektowania systemów a wykorzystaniem metod formalnych (zwłaszcza model checkingu) w procesie projektowania. Jako przykład, przedstawia sie pokrótce badania nad środowiskiem programowym COSMA, prowadzone w Instytucie Informatyki Politechniki Warszawskiej.

## 1 Wstęp

### 1.1 Metody formalne a proces projektowania

Jednym ze zjawisk, które dają się zaobserwować we współczesnej informatyce jest dysproporcja między rozwojem badań nad formalnymi metodami opisu i analizy systemów a stosunkowo słabym ich przenikaniem do praktyki projektowania.

Istotnie, w środowiskach naukowych i akademickich zainteresowanie badaniami w tej dziedzinie w ostatniej dekadzie wyraźnie wzrosło. Można ocenić, że co roku około dwudziestu międzynarodowych konferencji jest poświęconych w całości lub przynajmniej w znacznej części metodom formalnym. Dość wspomnieć periodyczne konferencje FME (*Formal Methods Europe*), FORTE (*IFIP International Conference on Formal Techniques for Networked and Distributed Systems*, czy multi-konferencję ETAPS (*European Joint Conferences on Theory and Practice of Software*, w 2003 roku w Warszawie). Wydawane są czasopisma naukowe, jak *Formal Methods in System Design* oraz *Formal Aspects of Computing Science*. Obok tych wyspecjalizowanych konferencji i czasopism, tematyka metod formalnych w projektowaniu pojawia się często w informatycznych czasopismach naukowych o szerszym światowym obiegu. Powstają wartościowe monografie na ten temat (np. [5, 2]) oraz narzędzia programowe, przewidziane do formalnej weryfikacji projektów sprzętowych i software'owych. Zainteresowany czytelnik znajdzie więcej danych na ten temat w witrynie [3].

Z drugiej strony, większość rezultatów i narzędzi z dziedziny metod formalnych jest mało znana poza środowiskiem naukowym i akademickim, chociaż wszyscy badacze starają się, by ich rezultaty miały charakter metod o wartości przemysłowej (*industrial strength formal methods*). Metody formalne z kręgu weryfikacji modelowej (*model checking*) są jeszcze stosunkowo często stosowane przy projektowaniu rozwiązań sprzętowych (*hardware*), jednakże praktyczna formalna weryfikacja projektów *software*'owych należy do rzadkości. Z tych zapewne powodów mówi się, że podczas gdy błąd w komercyjnie dostępnym układzie scalonym jest sensacją na światową skalę — błąd w równie powszechnym produkcie software'owym jest nie wartą uwagi codziennością.

Do najszerzej stosowanych w praktyce sposobów zwiększania zaufania co do poprawności projektów software'owych należą: testowanie i próbna eksploatacja rzeczywistej lub symulacyjnej wersji produktu. Jednakże, przy pomocy testów można sprawdzić, czy system w zasadzie *robi to, co powinien* robić, nie można natomiast sprawdzić, czy *nie robi tego, czego nie powinien* (np. zawieszać się). Również próbna, nawet długotrwała, eksploatacja systemu (zwłaszcza współbieżnego) nie gwarantuje, że akurat w jej trakcie pojawią się losowe koincydencje zdarzeń, prowadzące do ujawnienia się błędu synchronizacji czy koordynacji między współbieżnymi komponentami. Metody formalne usiłują *dowieść* istnienia (nieistnienia) błędu, nie zaś wytropić go eksperymentalnie.

Również komercyjnie oferowane narzędzia CASE i środowiska programistyczne, przewidziane do wspomagania produkcji software'u, praktycznie nie zajmują się formalną analizę poprawności zachowań projektowanych systemów. Koncentrują się raczej na składniowej poprawności software'u, zgodności jego konstrukcji z przyjętym paradygmatem programowania (np. strukturalnego czy obiektowego), a także na wspieraniu testowania i zarządzania samym procesem projektowania (praca zespołowa, kontrola wersji, dokumentacja itp.). Oferowane współcześnie sposoby specyfikacji zachowań systemu (np. UML [15]) są

ukierunkowane bardziej na doraźną wygode użytkownika, niż na formalną poprawność projektu. Jedynie nieliczne (i nie tak powszechnie używane) środowiska tego typu (jak np. EDT, oparte na zastosowaniu języka Estelle [17, 18]), oferują formalnie zdefiniowaną semantykę stosowanych konstrukcji programistycznych.

Znaczna część trudności ma z pewnością źródła w niedoskonałości samych metod i narzędzi do formalnej weryfikacji, tworzonych w środowiskach naukowych i akademickich. Niezależnie od tego, stosowanie ich w praktyce wymaga od projektanta po pierwsze opanowania nowego zestawu pojęć i samej 'technologii' weryfikacji, po drugie — opisywania zarówno systemu, jak właściwości podlegających weryfikacji — w formalny sposób wygodny dla danego narzędzia czy metody, lecz obcy dla projektanta-praktyka.

Tym zapewne można tłumaczyć małą praktyczną znajomość metod opartych np. na notacji Z, VDM, czy B-method, a także technik i narzędzi formalnych z kręgu dowodzenia twierdzeń (*theorem proving*), jak np PVS, HOL czy Larch, które wymagają sporej kultury matematycznej. Wiekszym zainteresowaniem zdają się cieszyć metody skończenie stanowe, w szczególności weryfikacja modelowa (*model checking*, [1, 2, 5, 8]). Jej krótkiej charakterystyce poświęcony jest sekcja 1.2.

### 1.2 Zasady model checkingu

W najogólniejszym zarysie, weryfikacja modelowa (model checking) polega na tym, że:

- zachowania komponentów systemu opisuje się w postaci pewnego typu automatów skończonych (o skończonej liczbie stanów i etykietowanych przejść między nimi),
- z opisu zbioru komponentów, przy odpowiednich założeniach co do zasad ich synchronizacji (np. model synchroniczny, asynchroniczny, przeplotowy, ...) otrzymuje się (oczywiście również skończony, choć może bardzo wielki) graf stanów (lub funkcję przejść) całego systemu, opisujący wszystkie możliwe zachowania modelu,
- właściwości podlegające ewaluacji są zadawane formalnie, zazwyczaj w postaci formuł pewnej logiki temporalnej (*temporal model checking*) lub automatu Büchiego.
- ewaluacja wymagań co do poprawności zachowań systemu polega na *wyczerpującej inspekcji* grafu stanów osiągalnych (lub funkcji przejść) systemu.

Podstawową zaletą tego podejścia jest fakt, że wszystkie pytania o zachowania systemu opisanego skończonym modelem są rozstrzygalne, jeśli odsunąć problem złożoności obliczeniowej algorytmów przeszukiwania grafu stanów systemu oraz algorytmu samego otrzymywania tego grafu. Co więcej, algorytmy inspekcji grafu, raz opracowane przez twórców metody — nie wymagają potem inwencji użytkownika, który musi jedynie nauczyć się formułować pytania i odczytać odpowiedź. Ważne, że w przypadku odpowiedzi negatywnej model skończenie-stanowy może dostarczyć kontrprzykładu, to znaczy wymienić ścieżkę w grafie, prowadzącą do miejsca (stanu lub podgrafu), w którym wymogi poprawności zostały naruszone. Analiza kontrprzykładów ułatwia identyfikację i usuwanie błędów.

Inna właściwość, predysponująca model checking do roli forpoczty metod formalnych w praktyce, polega na tym, że każdy komponent jest oddzielnym automatem, a opis jego zachowania (w kategoriach stanów i przejść) odpowiada intuicji projektanta i ułatwia późniejsze przekształcenie go sekwencyjny wątek, proces, automat sprzętowy itd. Tej właściwości nie mają np. systemy tranzycyjne czy sieci Petriego.

Z drugiej strony, modele skończenie-stanowe praktycznie nie nadają się (bez dodatkowych mechanizmów abstrakcji) do opisu abstrakcyjnych typów danych i operacji na nich: dopuszczają właściwie jedynie dane binarne i krótkie liczby całkowite. Źle sobie radzą również z innymi aspektami zachowań systemu, np. z dynamicznie zmienną liczbą procesów (wątków, klientów itp.) czy istnieniem buforów (kolejek) o nieograniczonej długości. Dlatego naturalnym i pierwotnym terenem zastosowania modeli skończenie-stanowych było projektowanie i weryfikacja układów sprzętowych ([4]) oraz *control-dominated systems*, to znaczy takich systemów, w których analizuje się poprawność sterowania i koordynacji między komponentami, nie zaś poprawność operacji na danych.

Problemem, z jakim walczą twórcy wszystkich metod i narzędzi skończenie stanowych jest *wykładnicza eksplozja modelu* wraz ze wzrostem liczby komponentów. Dlatego praktyczny sukces uwarunkowany jest opracowaniem skutecznych metod hierarchizacji opisu automatowego i redukcji modelu, a także metod badania modelu 'częściami' (*compositional model checking*).

Dla celów weryfikacji modelowej opracowano wiele narzędzi (model checkerów). Do najczęściej cytowanych w literaturze należą SPIN ([6], [7]), SMV ([8], [9]) i Cospan (lub jego komercyjna wersja FormalCheck, [10]). Kilkadziesiąt innych implementowano dla celów akademickich i badawczych ([3]).

Niżej zostanie opisane środowisko programowe COSMA ([16]) powstałe w Instytucie Informatyki Politechniki Warszawskiej w toku badań nad weryfikacją modelową. Jest ono oparte na modelu automatowym Concurrent State Machines (CSM, [25]). Model CSM wydaje się szczególnie dobrze nadawać do opisu i badania zachowań *współbieżnych systemów reaktywnych*, to znaczy takich systemów, w których szczególnie ważna jest wzajemna kooperacja zarówno między współbieżnie działającymi komponentami systemu, jak między systemem a jego otoczeniem. Pozwala on na wyrażenie dwóch form współbieżności: jednoczesnego występowania zdarzeń komunikacyjnych (formalnie: symboli alfabetu wejściowego) i jednoczesnego wykonywania przez komponenty swoich akcji. Nie zakłada się żadnego mechanizmu przeplatania akcji (*interleaving*) ani szeregowania symboli wejściowych automatu. Komponenty systemu są więc względem siebie asynchroniczne.

## 2 Automaty współbieżne CSM i środowisko COSMA

### 2.1 Automaty CSM

Technikę opisu zachowań systemu przy pomocy automatów współbieżnych CSM (Concurrent State Machines, [25]) opiszemy nieformalnie na przykładzie prostego systemu o schemacie strukturalnym jak na Rys. 1. Przyjmijmy, że Nadawca przygotowuje i stopniowo wkłada do bufora kolejne porcje danych (*put*). W chwili, gdy liczba danych w buforze osiągnie $N_1$ - Nadawca zatrzymuje się i czeka, aż Odbiorca przetworzy całą zawartość bufora, opróżni bufor (*reset*) i potwierdzi koniec swej pracy sygnałem potwierdzenia *ack*. Wówczas Nadawca znów podejmuje zapełnianie bufora itd. Załóżmy, że Odbiorca śledzi stan zapełnienia bufora i samoczynnie podejmuje przetwarzanie, gdy liczba porcji danych staje się równa $N_2$. Oczywiście, system działa poprawnie, jeśli $N_1 = N_2$. Dla zilustrowania konsekwencji błędnego zachowania systemu, rozpatrzymy jednak przypadek $N_1 > N_2$, tak, jak gdyby na skutek nieporozumienie między projektantami Nadawcy i Odbiorcy - Odbiorca rozpoczynał przetwarzanie zbyt wcześnie.

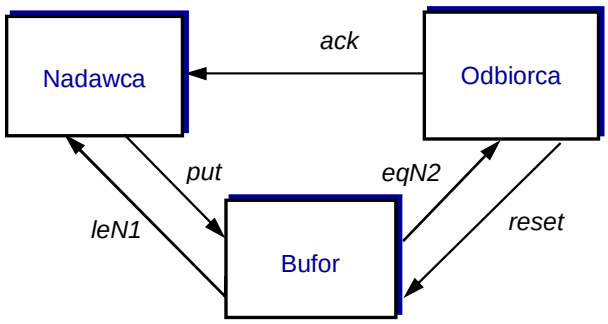



**Fig. 1.** Schemat strukturalny przykładowego systemu

W każdym momencie, Bufor (początkowo pusty) sygnalizuje Nadawcy, czy liczba danych jest mniejsza od $N_1$ (wyjście $leN1$), a Odbiorcy, czy liczba przebywających w nim porcji danych jest równa $N_2$ (wyjście $eqN2$). Otrzymując na swym wejściu symbol *put* (od Nadawcy) Bufor zwiększa swój stan, zaś symbol *reset* (od Odbiorcy) powoduje wyzerowanie bufora.

Zachowanie Nadawcy i Odbiorcy, opisane w postaci automatów współbieżnych CSM, jest przedstawione na Rys. 2. Modelem Nadawcy jest graf, składającym się z czterech węzłów (stanów) i siedmiu skierowanych krawędzi. *Think* jest stanem początkowym. W dolnej części prostokąta stanu widnieje zbiór symboli wyjściowych produkowanych w danym stanie. Tak więc np. w stanie *Write*, Nadawca produkuje jednoelementowy zbiór symboli $\{put\}$, w stanach $Think, Decide$ i $Wait$ produkowany zbiór symboli jest pusty, Odbiorca w stanie *Conclude* produkuje zbiór dwuelementowy: $\{reset, ack\}$. Symbole te są produkowane *jednocześnie* i *przez cały czas* przebywania w danym stanie.

Krawędzie grafu są etykietowane nie symbolami alfabetu wejściowego (jak to ma miejsce w 'klasycznych' automatach skończonych) lecz *boolowskimi formułami*. Argumentami formuł są symbole produkowane przez inne automaty ('wnętrze') systemu lub jego otoczenie ('zewnętrze'). Operatory $+, *, !$ oznaczają (odpowiednio) boolowską sumę, iloczyn i negację. Formuła oznacza warunek konieczny do tego, by dana krawędź była 'potencjalnie aktywna' (*enabled*). Tak więc na przykład formuła $!leN1$ na

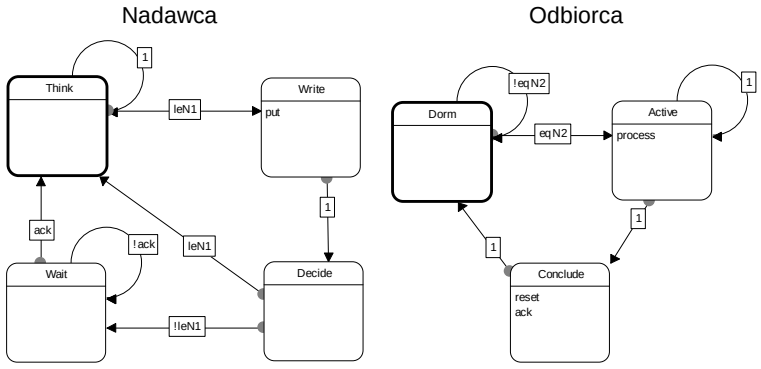



**Fig. 2.** Modele CSM Nadawcy i Odbiorcy

krawędzi od *Decide* do *Wait* oznacza, że Nadawca może przejść od stanu *Decide* do *Wait* jeśli *nie przyszedł* symbol $leN1$ itd.

Formuła **1** reprezentuje funkcję boolowska, która jest *true* niezależnie od aktualnego stanu wejścia maszyny. Krawędzie etykietowane przez **1** opisują zatem zachowania *spontaniczne*. Tak więc na przykład, wchodząc do stanu *Write*, Arbiter produkuje symbol wyjściowy *put*, a następnie spontanicznie przechodzi do stanu *Decide*. Formuła **0** (zawsze *false*) nie występuje w grafie, gdyż krawędzie, które nie są nigdy *enabled* są po prostu z grafu usuwane.

Jeśli dwie krawędzie wychodzące z tego samego stanu są jednocześnie *enabled*, dokonuje się wyboru jednej krawędzi aktywnej (*active*), która wskazuje następny stan. Zakłada się, że wybór jest niedeterministyczny i uczciwy (*fair*), co w praktyce znaczy, że każde potencjalnie aktywne przejście będzie kiedyś wykonane. Na przykład, Odbiorca w stanie *Active* może — niederministycznie i spontanicznie — albo pozostać w tym stanie, albo przejść do *Conclude*. Praktycznie znaczy to, że w stanie *Active* Odbiorca przebywa przez nieokreślenie długi, ale skończony czas[1]. Podobnie niedeterministyczne jest zachowanie Nadawcy w stanie *Think*.

W modelu CSM *przejściami* (tranzycjami) nazywane są jedynie krawędzie łączące dwa *różne* stany. Automat w każdej chwili przebywa więc w dokładnie jednym stanie, a przejścia są natychmiastowe (w zerowym czasie). Natomiast t. zw. 'ucha' (krawędzie od stanu z powrotem do tego samego stanu) opisują warunki *trwania* w danym stanie.

Model Bufora (Rys. 3) ilustruje pewną możliwość abstrakcji. Zbiór liczb całkowitych reprezentujących liczbę danych w Buforze można podzielić na pięć klas abstrakcji, odpowiadających stanom $B1 \ldots B5$. W każdym ze stanów produkowane są (lub nie) odpowiednie symbole $leN1$ lub $eqN2$. Warunkiem przejscia do 'wyższego' stanu jest pojawienie się symbolu *put* (od Nadawcy), symbol *reset* (od Odbiorcy) sprowadza Bufor do stanu $B1$.

Zakłada się, że komponenty systemu (tu: Nadawca, Bufor i Odbiorca) współistnieją we wspólnym środowisku komunikacyjnym. Rozpowszechnia ono do wszystkich komponentów (natychmiast i bezbłędnie) mnogościową sumę zarówno zbiorów symboli wyjściowych produkowanych przez wszystkie komponenty, jak zbiorów symboli przychodzących do systemu z jego zewnętrznego otoczenia[2]. Ta mnogościowa suma jest natychmiast odbierana przez wszystkie komponenty, które traktują ją jak aktualny zbiór swoich symboli wejściowych. Każdy z komponentów (asynchronicznie i niezależnie od innych) ewaluuje formuły boolowskie na krawędziach wychodzących z jego aktualnego stanu i zachowuje sie zgodnie z opisanymi wyżej zasadami. Nie nakłada się żadnego ograniczenia na jednoczesność występowania symboli (zdarzeń, sygnałów itd.) ani jednoczesność przejść (jak np. *interleaving*). Model CSM jest zatem współbieżny, asynchroniczny i koincydencyjny.

Globalne działanie całego systemu jest iloczynem (produktem) zachowań komponentów. Opracowano sprawny algorytm wyznaczania produktu automatów CSM [25], który prowadzi do otrzymania grafu stanów osiągalnych systemu . Operacja mnożenia ($\otimes$) jest przemienna i łączna. Produkt kilku komponentów jest z powrotem *pojedynczą* maszyną CSM. Ważne, że formuły boolowskie przypisane krawędziom produktu odwołują się wyłącznie do obecności (nieobecności) symboli przychodzących do

[1] W podstawowym modelu CSM nie specyfikuje się ograniczeń czasowych. 'Czasowa' wersja modelu, przewidziana do model checkingu, jest w trakcie opracowywania. Do celów *symulacji* (choć nie model checkingu) zachowania systemu w czasie opracowano rozszerzony model ECSM, patrz punkt 2.2

[2] Omawiany przykładowy system nie odbiera sygnałów z otoczenia

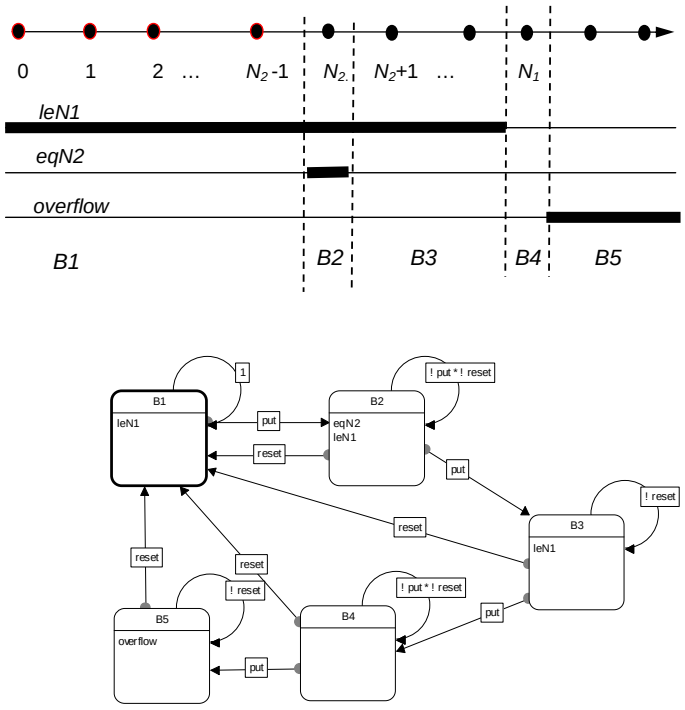



**Fig. 3.** Model CSM Bufora

systemu z zewnątrz, natomiast symbole, którymi komponenty komunikują się *wewnątrz* systemu — nie występują w tych formułach.

Produkt Nadawca ⊗ Bufor ⊗ Odbiorca (Rys. 4) ujawnia, iż system działa źle. Szarym kolorem zaznaczone są dwa stany zakleszczenia (*deadlock*), po osiągnięciu których system pozostaje bez możliwości wyjścia. Ponadto, w niektórych stanach symbole *put* i *process* występują jednocześnie, co oznacza, że zachowania Nadawcy i Odbiorcy są źle skoordynowane. Każda ścieżka w grafie stanów osiągalnych, prowadząca od stanu początkowego do takiego stanu niepożądanego jest tzw. kontrprzykładem (*counterexample*). Analiza kontrprzykładu pozwala na identyfikację (i w konsekwencji skorygowanie) błędu projektowego.

Powyższy przykład ma oczywiście charakter poglądowy. Jego celem było nieformalne wprowadzenie w tematykę weryfikacji z użyciem automatów CSM, a zwłaszcza zilustrowanie faktu, że zasady opisywania zachowań przy użyciu automatów CSM, interpretacja pojęć: system, produkt systemu itd. — wydają się naturalne i bliskie intuicji każdego, kto zna pojęcie grafu, stanu, przejścia, symbolu elementarnego i formuły boolowskiej. Graf z Rys. 4 ma 24 stany, można go narysować i zanalizować gołym okiem. W praktyce weryfikacji należy się liczyć z grafami osiągalności o rozmiarach setek tysięcy czy milionów stanów. Do tworzenia i analizy grafu stanów osiągalnych systemu są więc niezbędne odpowiednie algorytmy i narzędzia programowe. W przypadku modelu opartego na automatach CSM rolę tę odgrywa środowisko programowe COSMA.

## 2.2 Środowisko COSMA

Poglądowy schemat środowiska COSMA ([16]) jest przedstawiony na Rys. 5. Centralną jego część stanowi repozytorium, w którym (w postaci plików tekstowych w języku CXL) przechowywane są opisy modeli CSM. Język CXL został opracowany dla celów reprezentacji modeli CSM oraz ECSM w oparciu o XML. Funkcje sterowania całością narzędzia pełni (nie pokazany na schemacie) nadrzędny moduł sterujący, który umożliwia:

- utworzenie i nazwanie aktualnej przestrzeni roboczej (workspace),
- tworzenie, usuwanie i edycję projektów w tej przestrzeni,
- wywoływanie pozostałych modułów i komunikację między nimi.

Funkcje pozostałych podstawowych modułów są następujące:

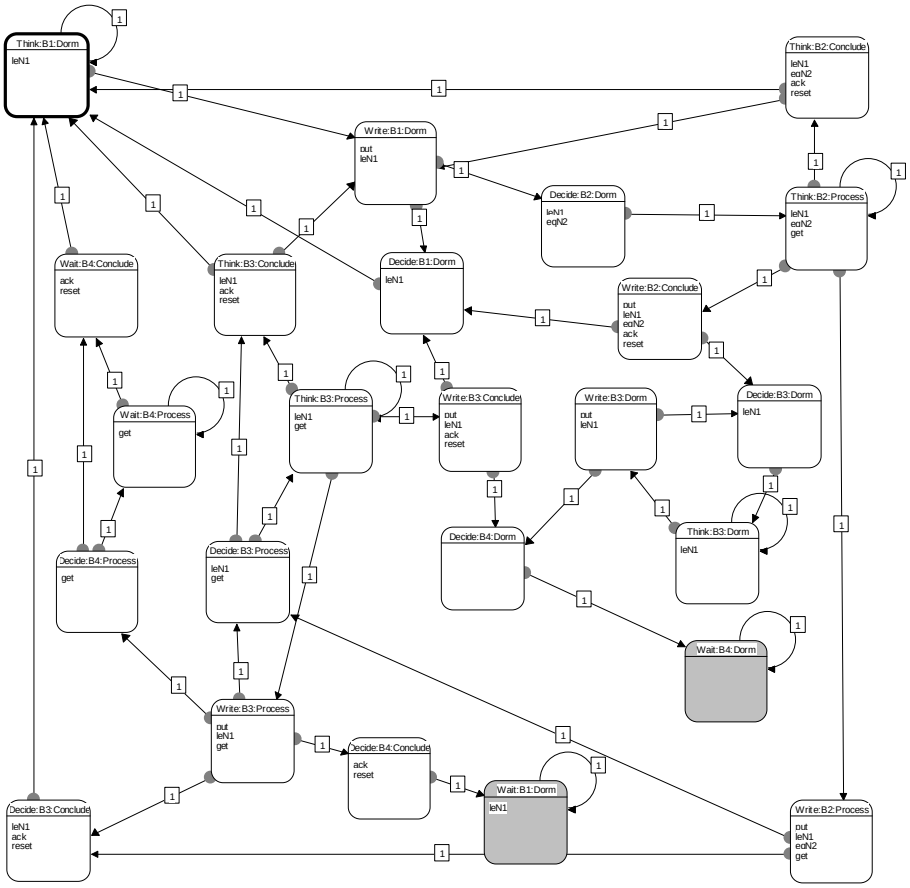



**Fig. 4.** Graf stanów osiągalnych systemu (Produkt Nadawca ⊗ Bufor ⊗ Odbiorca )

- *Grapher* umożliwia projektantowi (graficzne) tworzenie i edycję modeli CSM komponentów systemu. Modele są zapamiętywane w formacie CXL lub eksportowane do formatu emf.
- *Product Engine* tworzy reprezentację komponentów w postaci BDD i wykonuje operację mnożenia (⊗) komponentów tworzących jeden projekt. Wynik ma postać BDD, może być także przetworzony na postać tekstową CXL,
- *TempoRG* przyjmuje od projektanta wymagania wyrażone w postaci wyrażeń logiki temporalnej QsCTL [23] [26] i przeprowadza ich ewaluację w grafie stanów osiągalnych (produkcie) dostarczonym przez *Product Engine*,
- *Cntrexmple Editor* produkuje w postaci graficznej lub tekstowej kontrprzykłady, dostarczane przez *TempoRG* w przypadku negatywnego rezultatu ewaluacji,
- *Reduction Engine* przeprowadza redukcję wskazanego produktu, dostarczonego przez *Product Engine* lub pobranego z repozytorium.

Narzędzie COSMA wspiera również rozszerzony model CSM (Extended CSM, ECSM, [24]). Nie jest on modelem skończenie-stanowym i nie podlega model-checkingowi, jest natomiast wykonywany w drodze symulacji i może służyć do ewaluacji wydajności systemu (moduł *ECSM Simulator*, ([22]). Rozszerzenia w stosunku do modelu CSM są następujące:

- istnieje możliwość definiowania dowolnych zmiennych lokalnych dla komponentu lub globalnych dla projektu,
- zarówno stanom, jak przejściom komponentów CSM można przyporządkowywać akcje, będące po prostu sekwencyjnymi programami w C,
- formułom boolowskim na krawędziach grafów CSM można dodawać czynniki odpowiadające wyrażeniom boolowskim na zmiennych,
- symbole generowane w stanach mogą być uzupełnione o atrybuty (modelowanie zawartości komunikatów),

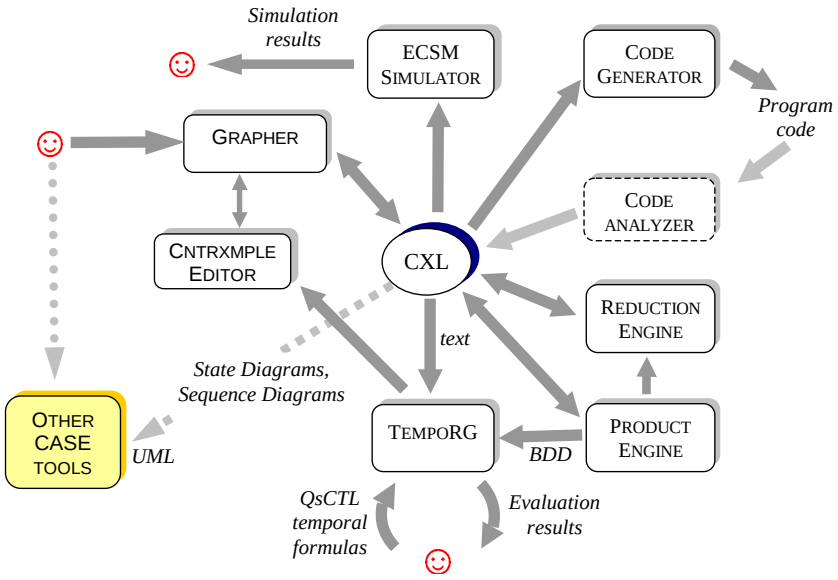



**Fig. 5.** Ideowy schemat środowiska programowego COSMA

– dla celów badań wydajnościowych, stanom można przypisywać czasy (deterministyczne lub losowe czasy przebywania w stanie), zaś rozejściom niedeterministycznym - prawdopodobieństwa.

Do modułów środowiska COSMA należy również *Code Generator* — generator kodu programu z modeli ECSM, istniejący w nieopublikowanej wersji eksperymentalnej. Moduł *Code Analyzer* był z kolei planowany jako narzędzie produkujące modele CSM komponentów zapisanych w postaci programów w jęz. Java. Wejściem dla modułu jest nie sam program, lecz jego odpowiednik w postaci pośredniej (t. zw. BIR) produkowanej przez środowisko Bandera ([13], [14]). Eksperymentalna wersja tego modułu [21], choć działa, nie zachęca jednak do kontynuacji: wydaje się, że skończenie stanowe modele można skutecznie produkować dla bardzo ograniczonego podzbioru Javy.

## 3 Podsumowanie

Próba przerzucenia pomostu między praktyką projektowania a formalną weryfikacją musi wiązać się przede wszystkim z określeniem miejsca formalnej weryfikacji w procesie projektowania. Dotychczasowe doświadczenia skłaniają do poglądu, że zalety model checkingu ujawniają się najlepiej na wczesnym etapie projektowania. Analiza koncepcyjnego szkieletu systemu, uwzględniajacego koordynację i komunikację między głównymi współbieżnymi komponentami może ujawnić istnienie błędów komunikacji i synchronizacji, które (po usunięciu) nie będą propagowały się do dalszych etapów projektowania. Tak zweryfikowane automatowe modele komponentów mogą następnie posłużyć jako wzorce do szczegółowej implementacji w postaci programów czy układów sprzętowych.

Ze względu na pożądaną 'przyjazność' metodologii, w pracach nad narzędziami do model checkingu należałoby przywiązywać większą wagę do ich współpracy z powszechnie znanymi narzędziami CASE i środowiskami programowania. W szczególności, warto wykorzystać podobieństwa między modelami skończenie stanowymi a diagramami stanów, sekwencji, aktywności i kooperacji (UML) lub statecharts Harela [11], [12]. Eksport i import tego typu specyfikacji między narzędziem CASE a model checkerem takim, jak COSMA, przybliżyłby możliwość formalnej weryfikacji projektu na wczesnym etapie projektowania.

Automaty współbieżne CSM i środowisko COSMA wydają się obiecujące jako koncepcyjna i narzędziowa podstawa tego typu prac. Przemawiają za tym:

– koncepcyjna prostota modelu komponentów systemu,
– graficzny tryb definiowania zachowań komponentów (zgodnie ze współczesnymi tendencjami, por. choćby UML),
– proste zasady komunikacji wewnątrz systemu (asynchroniczność, jednoczesność, brak domniemanego 'demona' przeplatającego lub szeregującego zdarzenia i akcje),

– istnienie podstawowych narzędzi do otrzymywania i analizy grafu stanów osiągalnych systemu, a także symulacji modelu ESCM.

Oczywiście, konieczny jest dalszy rozwój metodologii, zwłaszcza zaś:

– opracowanie metod hierarchicznej specyfikacji automatów CSM,
– wsparcie dla algorytmów eksportu/importu automatów CSM z/do UMLowych diagramów stanów, sekwencji, aktywności i kooperacji,
– wprowadzenie ograniczeń czasowych do modelu CSM (Timed CSM),
– modelowanie operacji na danych,
– opracowanie specyficznych dla CSM algorytmów redukcji, pozwalających złagodzić efekt wykładniczej eksplozji modelu,

Tak ukierunkowane badania są w toku. Dotychczasowe rezultaty są zachęcające [28, 29, 27]. Niezależnie jednak od ich wartości poznawczej, do szerszego stosowania opisanej metodologii w praktyce najbardziej zachęciłyby przykłady, *case studies*, które mają największą siłę przekonywania.

## References


1. E. M. Clarke, O. Grumberg, D. A. Peled: *Model Checking*, MIT Press, 2000.
2. B. Berard (ed.) et al.: *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer Verlag, 2001,
3. `http://archive.museophile.sbu.ac.uk/formal-methods.html`
4. T. Kropf: *Introduction to Formal Hardware Verification*, Springer Verlag, 1999.
5. D. A. Peled: *Software Reliability Methods*, Springer Verlag, 2001,
6. G. J. Holzmann: *The Model Checker SPIN,* IEEE Trans. on SE, Vol. **23,** No. 5 (May 1997), p. 279–295.
7. *SPIN*: http://spinroot.com/spin/whatispin.html
8. K. L. McMillan: *Symbolic Model Checking,* Kluwer Academic Publishers, 1993.
9. *SMV*: `http://www-2.cs.cmu.edu/ modelcheck/smv.html`
10. *FormalCheck*: `www.cadence.com/datasheets/formalcheck.html`
11. Harel D., *StateCharts: A visual formalism for complex systems,* Science of Computer Programming, **8,** p. 231–274, 1987.
12. Harel D. et al.: *STATEMATE: A Working Environment for the Development of Complex Reactive Systems,* IEEE Transactions on Software Engineering, **16** (4), p. 403–414, April 1990.
13. J. Hatcliff, M. Dwyer: *Using the Bandera tool set to model-check properties of concurrent Java software,* Proc. CONCUR 2001, 2001, pp. 39–58.
14. Bandera: `www.cis.ksu.edu/santos/bandera`
15. *Unified Modeling Language,* `http://www.rational.com/uml`
16. *COSMA,* `www.ii.pw.edu.pl/cosma/`
17. S.Budkowski et al.: *The Estelle Development Toolset,* Institut National des Télécommunications, Evry, France, 1998, `http://www-lor.int-evry.fr/edt`
18. *Information Processing Systems. Estelle: A Formal Description Techinque Based on an Extended State Transition Model*, ISO/TC97/SC21, 1997.
19. W. B. Daszczuk: *Evaluation of temporal formulas based on checking by spheres,* Proc. Euromicro Symposium on Digital Systems Design—Architectures, Methods and Tools, IEEE Computer Society, September 4-6, 2001, Warsaw, pp. 158—164.
20. W. B Daszczuk, W. Grabski, J. Mieścicki, J. Wytrębowicz J.: *System modeling in the COSMA environment,* Proc. Euromicro Symposium on Digital Systems Design - Architectures, Methods and Tools, IEEE Computer Society, September 4–6, 2001, Warsaw, pp. 152–157.
21. P. Fusik: *Weryfikacja modelowa współbieżnych programów w języku Java przy użyciu środowiska Bandera i COSMA,* Praca dyplomowa magisterska, Instytut Informatyki PW, 2004.
22. A. Krystosik: *Projektowanie i formalna weryfikacja oprogramowania dla reaktywnych systemów ochrony informacji przy pomocy automatów ECSM,* Materiały z VI Krajowej Konferencji Zastosowań Kryptografii Enigma 2002, Warszawa, 15–17 maja 2002, str. 185–193.
23. W. B. Daszczuk: *Verification of temporal properties in concurrent systems*, Rozprawa doktorska, Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Warszawa, styczeń 2003,
24. A. Krystosik: *ECSM—Extended Concurrent State Machines,* ICS Research Report 2/2003.
25. J. Mieścicki: *Concurrent State Machines, the formal framework for model-checkable systems,* ICS Research Report, 5/2003,
26. W. B. Daszczuk: *Temporal model checking in the COSMA environment (the operation of TempoRG program),* ICS Research Report, 7/2003, Warszawa, 2003.
27. W. B. Daszczuk: *Timed Concurrent State Machines,* ICS Research Report, 27/2003, Warszawa, 2003.
28. J. Mieścicki: *Multi-phase model checking in the COSMA environment,* ICS Research Report, 14/2003, Warszawa, 2003.
29. J. Mieścicki, B. Czejdo, W. B. Daszczuk: *Multi-phase model checking in the COSMA environment as a support for the design of pipelined processing,* ICS Research Report, 16/2003, Warszawa, 2003.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.195-203
ISBN 83-922646-0-6


# Formal Verification of a Three-stage Pipeline in the COSMA Environment[1]

**Jerzy Mieścicki**, **Wiktor B. Daszczuk**

Institute of Computer Science,
Warsaw University of Technology, ul. Nowowiejska 15/19,
PL 00-665 Warsaw, Poland.
{J.Miescicki, W.Daszczuk}@ii.pw.edu.pl

**Abstract.** The case study analyzed in the paper illustrates the example of model checking in the COSMA environment. The system itself is a three-stage pipeline consisting of mutually concurrent modules which also compete for a shared resource. System components are specified in terms of Concurrent State Machines (CSM) The paper shows verification of behavioral properties, model reduction technique, analysis of counter-example and checking of real time properties.

## 1 Introduction

In [1] we have described the functional model of a system for processing of consecutive portions of data (or messages) submitted to its input. Each message goes through the three stages of processing which is reflected in the system's structure (Fig. 1). The system is a three–stage pipeline consisting of three modules that operate concurrently and asynchronously, in a sense that there is no general, common synchronizing process or mechanism. Moreover, two out of three modules compete for the access to the common resource, which is accessed also by some other (unspecified) agents from the system's environment. This calls for the verification if the cooperation among system components is *correct*. Indeed, due to potential coordination errors system can get deadlocked, messages can be lost or duplicated etc.

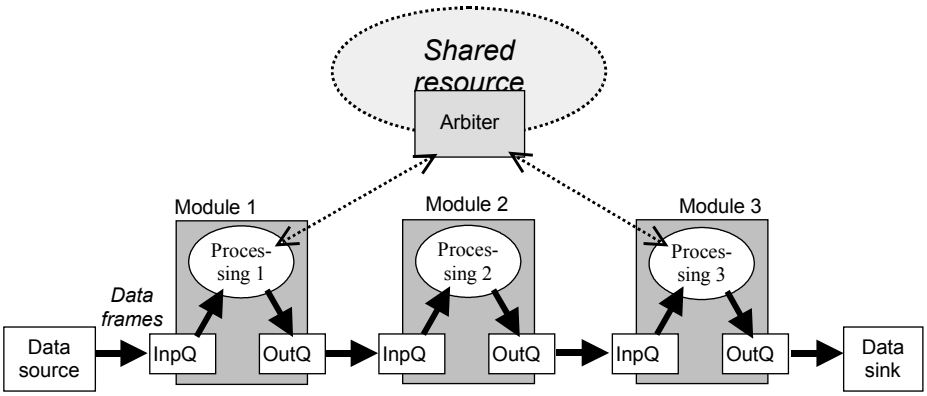



**Fig. 1.** Flow of data in a three-module pipeline with a shared resource

It is known that in the case of asynchronous and concurrent systems such errors are extremely hard to discover, identify and correct using typical debugging and testing procedures. Therefore, we have applied a formal procedure of *model checking* [2, 3, 4, 5], using the software tool called COSMA [6], implemented in the Institute of Computer Science, Warsaw University of Technology.

Model checking is based on the following principle. Given the finite state model *M* of system's behavior and property (requirement) *p* to be checked, one has to check if *p holds for M*. Usually, there is a set of techniques and algorithms (making together the model-checking environment or tool) designed for this purpose. This is

1 This work has been supported by grant No.7 T 11 C 013 20 from Polish State Committee for Scientific Research (Komitet Badań Naukowych).

designer's job to formulate properties to be evaluated: usually the verification involves a set of model checking experiments with several properties $p_i$. Additionally, if the given property does not hold for *M*, then so-called counterexample is provided which allows to identify the sequence of states (or events) that results in this negative evaluation. This helps to identify and correct the cause of an error.

The main limitation the model checking faces is the *exponential explosion* of model's state space size along with the increase of the number of finite state system components and their individual state spaces. So, an extensive research is being done on various techniques that can help to manage the problem. First, multiple forms of *reduction* of state space are proposed, aimed to remove the states and transitions which are irrelevant w.r.t. the evaluation of a given formula. The other approach is to calculate the state space just during the evaluation, as one can expect that in order to obtain the outcome of the evaluation only the *bounded model* will do. Still other technique consists in *compositional model checking*, where some individual parts of a system (of more acceptable size) are subject to an exhaustive state space search while the conclusion as to the behavior of a whole system is reached by some logical reasoning. Unfortunately, most of ideas of reduction found in the literature (e.g. [7, 8]) usually cannot be applied for Concurrent State Machines. This model admits *coincident* execution of actions rather than their interleaving, while most finite state models assume just the interleaved executions. Also, other known forms of reduction (e.g. *slicing* and *abstraction* [9, 10, 11]) make the use of specific properties of programs and can not be applied directly to more abstract CSM models.

In this paper we briefly describe three techniques used in a COSMA-style methodology of system's verification. First, we will analyze the system behavior step-by-step, using so-called multi-phase reduction [12, 13] which exploits some compositional features of the CSM model [14]. As a result, the system which (as naively estimated) may have as much as $4*10^{14}$ states is finally reduced to a model of 323 states and 1406 edges, easily representable and algorithmically checkable in a split second. Then, as some properties proved to be evaluated to *false*, we illustrate how the counterexample can be obtained and analyzed. Finally, using timed version of the model, we present how real time dependencies may be analyzed. This part of the COSMA environment is under development.

## 2 Two-phase procedure of obtaining the reduced reachability graph

Let us recall the basic facts about the CSM model of a pipeline, described in more details in [1]. It consists of three complex modules and three individual components (data source, data sink and the arbiter) common to the whole system. Each module can be internally subdivided into six components (see also left-hand part of Fig. 2 below). In total, this makes a set of 21 cooperating components. For each of them, a separate (finite state) CSM model has been developed, aimed to specify its behavior as well as the communication to/from its communication partners. The goal was to obtain the large system's behavioral model or a graph of reachable system states, containing all the reachable states and possible execution paths among them. Then, some temporal formulas representing desirable behavioral properties of the system have been evaluated (*true* or *false*).

In [1] the emphasis was put on the specification of components and temporal properties, while the technique of obtaining the product of all the components was not analyzed. Now we proceed to the method of determining the system's behavioral model that can (to an extent) help to cope with problems of the graph size. The main idea is devoted to is the following. In order to obtain system behavioral model, one has to perform the product ($\otimes$) of CSM models of system components. This operation is associative and commutative. Associativity supports the important compositional property. Now, if we have - for instance - system $Z = \{m, n, p\}$ of three components, then (due to the associativity) we can obtain the behavioral model either immediately, as a 'flat' product $\otimes Z = m \otimes n \otimes p$ or in two steps: first computing the local product of some subsystem e.g. $r = m \otimes n$, then $\otimes Z = r \otimes p$. Meanwhile, before the second, final product is obtained, we can apply some reduction procedure to the partial product *r*. While the associativity of the product applies to other finite state models as well, this reduction makes the use of intrinsic features of CSM model itself. If machines *m* and *n* do communicate intensively with each other - it may result in a considerable reduction of a total computational effort, necessary for the computation of $\otimes Z$.

Below, we show how this general rule apply to our system of 21 components, briefly recalled above. We will proceed in the two steps or *phases*.

Generally, each phase consists in the selection of some subsystem, obtaining its CSM product and removing the *irrelevant* states and edges from it. However, one has to decide first which elements of the model are *relevant* ones and therefore have to be preserved. Relevant – in this sense - are the selected *output symbols* (produced by individual system components) and thus also the system states in which these symbols are generated. Typically, among relevant symbols are:

1. symbols that are referred to in the temporal formulas to be evaluated,

2. symbols that should be preserved for designer's convenience, e.g. because they serve as a type of markers that make the complex behavior more readable,
3. symbols that are necessary from the viewpoint of the communication among the currently reduced subsystem and remaining components[2].

The former two groups of symbols are decided upon by the designer while the latter one is determined by the specification of system components. Assume that in our case the relevant symbols of the types 1 and 2 above are the following ones:

*msg_1, msg_4, doProc_1, doProc_2, doProc_3*

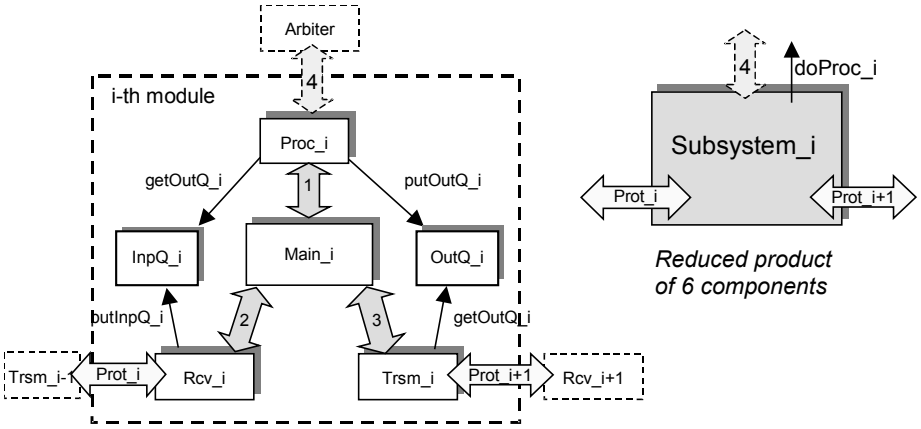



**Fig. 2.** 'Local' product of a single module

In order to obtain the 'phase-1' model of our example system we perform the following procedure:

Phase-1

1. Take a subsystem, consisting of the six components of module #1 (see Fig. 2),
2. Compute its CSM product,
3. Reduce it, leaving as the relevant output symbols the following ones:
   - All the output symbols (from the subsystem) which are 'watched for' by subsystem's communication partners (i.e. the *Arbiter*, *Trsm_0*, *Rcv_2*),
   - Symbols from the set selected above, which are produced within module #1 (this case: *doProc_1*),

Let the product reduced this way be called *Subsystem_1*,

4. Repeat the above for modules #2 and #3 obtaining *Subsystem_2* and *Subsystem_3* (respectively).
5. Substitute subsystems 1, 2, 3 in the place of just processed components.

This way we obtain the phase-1 structural model, in which subsystems 1, 2 and 3 are replaced by single automata. It is noteworthy that for *Subsystem_1*:

- Cartesian product of its six components has 24300 states,
- CSM product (before reduction) has 24 reachable states and 31 edges,
- After reduction, *Subsystem_1* is a graph of 10 states and 16 edges.

2 Note that among the 'remaining components' can be also the additional, auxiliary automata (e.g. *Invariant*, in our case) necessary for expressing the properties under checking.

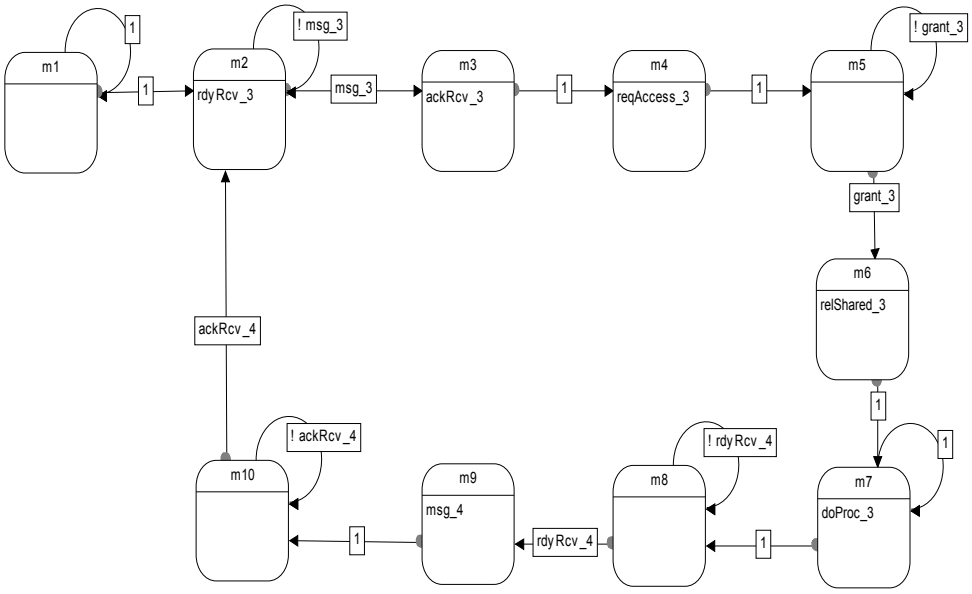



**Fig. 3.** CSM model of *Subsystem_3* (*reduced* product of six components of module #3)

For *Subsystem_3*, the situation is analogous. As an illustration, the reduced CSM product of six components making *Subsystem_3* (*Main_3, Rcv_3, Trsm_3, Proc_3, InpQ_3, OutQ_3*) is shown in Fig. 3. At no surprise, it has 10 states and 16 edges, same as *Subsystem_1*. For *Subsystem_2* (not shown), it is as few as 7 states and 12 edges.

Now, the analogous procedure can be applied again to the structural elements of the reduced model, for instance:

Phase-2

- The procedure is applied to a subsystem consisting of *Subsystem_1* and *Trsm_0*, preserving all the output symbols which are 'watched for' by the communication partners (i.e. *Arbiter* and *Subsystem_2*) and symbols needed for temporal formulas to be evaluated (this case: *doProc_1* and *msg_1*),
- and to a subsystem consisting of *Subsystem_3* and *Rcv_4*, preserving 'watched for' symbols (i.e. *Arbiter* and *Subsystem_2*) and symbols for model checking (this case: *doProc_3* and *msg_4*),
- finally substitute *Syst_1_Trsm_0* and *Syst_3_Rcv_4* in the place of just processed components.

This way we obtain the phase-2 structural model as in Fig. 4. Notice that the phase-2 system now consists of *four* components (instead of 21 components of phase-0 structural model), each of significantly reduced size. This 'downsizing' the model can be continued, but each time the reduction is performed a certain conditions have to be met [12] so that the reduction is not necessarily guaranteed. Nevertheless, in practice the degree of reduction can be substantial.

Let the CSM product of the system from Fig. 4 be called *New_System* and serve as the new behavioral model in which the temporal requirements are evaluated. *New_System*, obtained again with the COSMA Product Engine, has 323 states and 1406 edges and is expected to preserve at least these functional properties of the original, flat version which can be expressed in terms of symbols *msg_1, msg_4, doProc_1, doProc_2, doProc_3*.

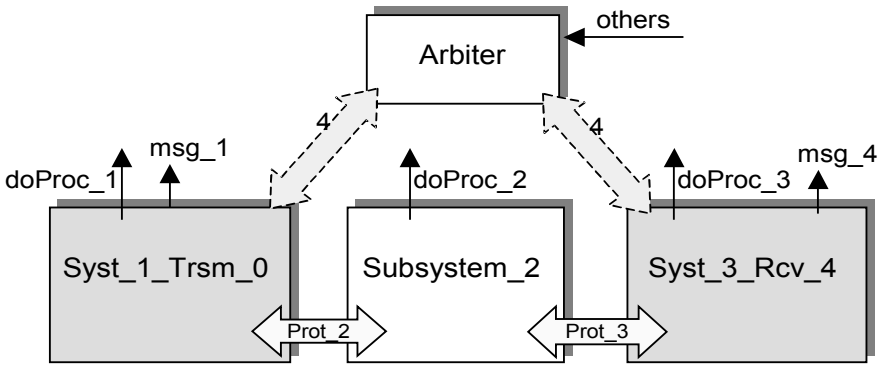



**Fig. 4. Phase-2 structural model of the system**

## 3 Verification of the reduced model

To sum up, now we have two behavioral models of the same example system:

- *Flat-product* (CSM product of 21 components, obtained as described in [1]) which had 8284 states and 34711 edges,
- *New_System*, obtained in the above two-phase reduction procedure, with 323 states, 1406 edges.

Both models have been verified in COSMA environment. As in [1], an additional automaton *Invariant* was determined to conveniently specify the verified properties. The checked properties were the following:

- Safety 1, saying – informally – that the number of messages within the pipeline never exceeds its capacity and the number of messages leaving the pipeline never exceeds the number of messages entering it,
- Liveness 1, saying – informally – that for any system state it is possible that the pipeline *eventually* would get empty,
- Liveness 2, saying – informally – that for any system state it is possible that the pipeline *eventually* would get full,

Experiments have been performed on PC computer with 800MHz processor and 512 MB RAM. Results are summarized in Table 1.

Notice that both formulas referring to the liveness have been evaluated *false* in both (flat and reduced) models. This negative result means that the system may enter such a state (states) that - from this state on - the pipeline is never empty again (i.e. it never terminates the processing of messages) or is never able to process three messages at once, which it was designed for.

The differences in evaluation times are really noteworthy: in all cases the ratio of $10^3$ – $10^4$ in favor of reduced model was achieved, even though in terms of state space size the reduced model is only approximately 25 times less than the flat one.

**Table 1.** Summary of experiments

| | Flat model | | Reduced model | |
|---|---|---|---|---|
| | Result | Evaluation time | Result | Evaluation time |
| Safety 1 | True | 17 s. | true | < 1 ms |
| Liveness 1 | False | 54 s. | false | < 1 ms |
| Liveness 2 | False | 4 min 40 s. | false | 60 ms |

Finally, the model verification can be summarized as follows;

- The system itself performs wrong: there must be a synchronization bug in the specification of components. This calls for the analysis of a counterexample,
- The reduced model well preserves the relevant properties of the primary, flat one. Indeed, each case the same temporal formula was evaluated the same way (*true* or *false*) in both models,
- The multi-phase reduction method provides a significant gain in the evaluation time, even greater than the savings in the state space itself.
- The advantages of the evaluation algorithm used in the COSMA tool have been also confirmed. The algorithm terminates the evaluation as soon as the result (*true*, *false*) is certainly determined. It is why the evaluation times of rather similar formulas (1) and (2) differ by a few dozen of times.

## 4 Analysis of a counterexample

In the case of negative evaluation, the TempoRG checker [15, 17] produces a counterexample. Often, it is a path (a sequence of states) in the reachability graph that leads to the state where it was decided that the temporal formula is to be certainly evaluated false. In the case of more complex temporal formulas involving several operators, the counterexample can be a tree [15], showing which particular part of the formula (a sub-formula) is responsible for the negative result. Tracing the consecutive states along the counterexample, the designer is able to identify the synchronization bug.

However, in the case of *reduced* models, the model states can be unreadable. As a result of reduction, some states are eliminated, remaining ones are usually renamed etc., so that the analysis of counterexample should be based on the sequences of symbols (events) produced by the system instead of on sequences of states.

The evaluation of both formulas representing the liveness condition yields the same counterexample, presented in Fig. 5. The counterexample itself pretends to be a CSM, in order to enable the use of animation feature of COSMA tool. Using it, one can trace the states of individual components (and their change) corresponding to consecutive states of an counterexample. Also, some additional symbols (not used in 'regular' CSM) are also introduced as first elements of states' output field. @ marks the starting state of the formula (this case it is system's initial state) while F and G stand for operators of sub-formulas (*G* stands for ***AG*** and *F* stands for ***AF***).

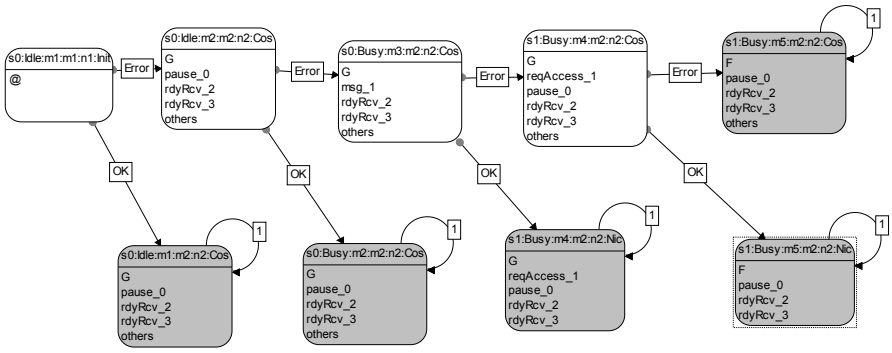



**Fig. 5.** Counterexample to the formula ***AG AF in*** *Invariant.s3*

The counterexample is constructed as follows:

- it begins in the starting state of evaluation (the initial state in this example),
- it contains sub-paths responsible for sub-formulas (which may produce a tree-like counterexample),
- for four states two successors are shown: one which leads towards an erroneous state (in which the error is possible, transition labeled with *Error*), the other which leads to a 'proper' state (transition labeled with *OK* ),
- the fifth state in an upper sequence, namely *s1:Busy:m5:m2:n2:Cos* is referred to as a *Trap.* The rules of constructing counterexamples [24] say that this state is a representative of so-called Ending Strongly Connected Subgraph (ESCS) of states in which the most nested formula (***in*** *Invariant.state*; *state*∈ *{s0,s3}*) is not satisfied. When the system falls into one of these states, the error is inevitable (the desired state *state* of *Iinvariant* is never reached).

The *analysis* starts with finding the last one of states (in the sequence) that has two outgoing transitions: one labeled *OK* and the other labeled *Error*. This state is referred to as a *Checkpoint*. In the example, it is (*s1:Busy:m4:m2:n2:Cos*), with two successors: (*s1:Busy:m5:m2:n2:Nic*) as 'proper' state and

(*s1:Busy:m5:m2:n2:Nic*) as a 'wrong' one. This time, the 'wrong' state is actually the *Trap* itself, but often it is only the initial state of a sequence of states which *inevitably* ends in a trap. Analysis of signals generated in the triangle *{Checkpoint, its 'proper' successor, its 'wrong' successor}* reveals the nature of error. We see that in the *Checkpoint* a request of access to the shared resource is generated (signal *req_Access_1*), and the resource is granted to another user (signal *others* is present). For this state, its 'proper' successor does not produce *others*, while in the *Trap* the symbol *others* is still present. So, *OK*-labeled transition (to a 'proper' state) is executed only if signal *others* is withdrawn, otherwise the system chooses a transition to a 'wrong' state which leads to the *Trap*. In other words, the error is inevitable, if the request (*req_Access_1*) is issued while other users do use the shared resource.

Actually, in the system the two-state dead-end subgraph (causing a livelock of the whole system) can be found. System performs incorrectly because *reqAccess_1* is not stored. Recall that in the CSM framework no implicit buffering of events is assumed: this should be provided by the model itself, e.g. by an additional (e.g. two-state) buffering component or by a simple modification of *Proc_1*. The same conclusion refers to the third module which accesses the shared resource as well. Both modules (#1 and #3) have been easily corrected and positively verified.

Finally, we may add that the flat product of the *corrected* system has 8086 states, 33588 edges instead of 8284 states and 34711 edges of the (incorrect) flat product discussed in [1]. This confirms the observation that the better the synchronization is, the less is the behavioral model of a system.

## 5 Real-time dependencies

Now we may convert automata to TSCM (Timed CSM, derived from CSM as Timed Automata [18, 19]) by adding time constraints and clock resets on some transitions in automata *Proc_i* and control units *Main_i* (Fig. 2). All time dependencies are shown as multiples of a basic time period, a *tick*. The constraints in *Proc_i* (Fig. 6) informs what is the minimal time of processing (*tim1*: by the constraint on the transition outgoing from the state *UseShared*) and the maximal time (*tim2*: the constraint on self-loop of the state *UseShared*). The constraints are based on a clock *Ti* local to *Proc_i*. The fixed time of staying in states in *Main_i* models delays in control unit. The clock is being reset every time the automaton enters *UseShared*. The constraints guarantee that the time of using a shared resource is finite. The constants *tim1_i* and *tim2_i*, *tim1_i* < *tim2_i*, may be specific to subsystems 1,2 and 3. Auxiliary automaton which guarantees finite time of using the resource by *others* must be modeled. Also, maximal time of a time period between generation of items should be specified.

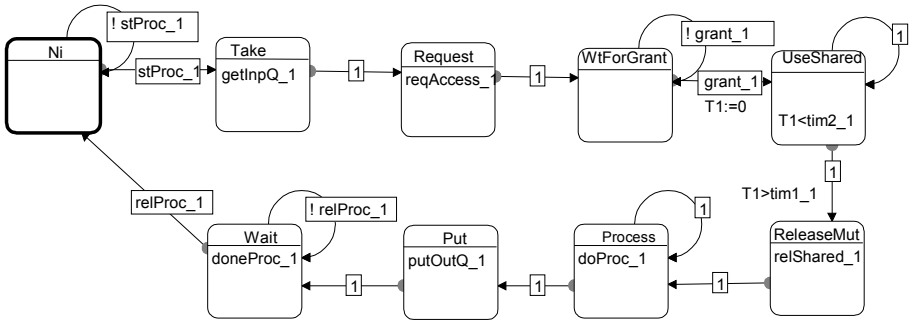



**Fig. 6.** Timed automaton *Proc_1*

Now we may convert automata to TSCM (Timed CSM, derived from CSM as Timed Automata [18, 19]) by adding time constraints and clock resets on some transitions in automata *Proc_i* and control units *Main_i* (Fig. 2). All time dependencies are shown as multiples of a basic time period, a *tick*. The constraints in *Proc_i* (Fig. 6) informs what is the minimal time of processing (*tim1*: by the constraint on the transition outgoing from the state *UseShared*) and the maximal time (*tim2*: the constraint on self-loop of the state *UseShared*). The constraints are based on a clock *Ti* local to *Proc_i*. The fixed time of staying in states in *Main_i* models delays in control unit. The clock is being reset every time the automaton enters *UseShared*. The constraints guarantee that the time of using a shared resource is finite. The constants *tim1_i* and *tim2_i*, *tim1_i* < *tim2_i*, may be specific to subsystems 1,2 and 3.

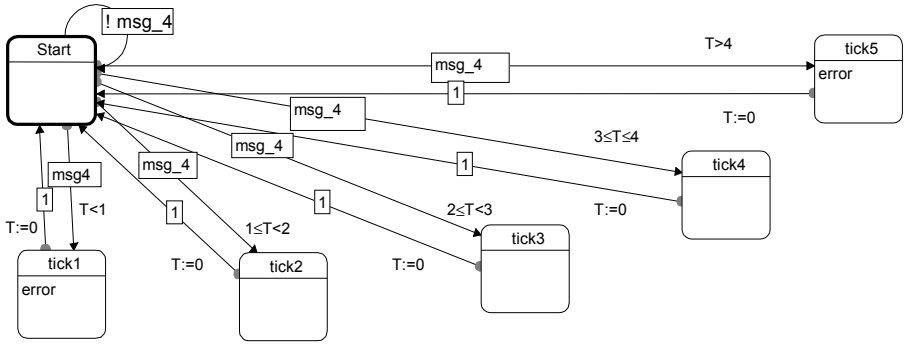



**Fig.7.** Testing timed automaton

Auxiliary automaton which guarantees finite time of using the resource by *others* must be modeled. Also, maximal time of a time period between generation of items should be specified. The model may be reduced as earlier, but in states subject to reduction their time constraints should match.

Now, the state space may be calculated following the rules given in [20], and RCSM (Region automaton) may be constructed from product TCSM (TCSM does not specify the succession relation unambiguously). Basing on RCSM state space, a testing automaton may be constructed, as shown in Fig. 7. This automaton checks if a time period between two items on output of the whole system is <0,1), <1,2), <2,3) ... ticks. If we impose a minimal and maximal time on the system, states violating the limits should produce the *error* signal (period <1 or >4 in this case).

The presented verification should be completed by a test using *invariant* automaton, because time constraints may change the behavior of the system.

## 6 Conclusions

The advantage of the formalism of Concurrent State Machines is that in order to understand (or even to design) the behavioral specification of a system component one has to be familiar with only a few elementary notions: a state, a transition, an atomic symbol, a Boolean formula, a time constraint. Generally, the semantics of an individual CSM is not far from the conventional finite state machines or basic UML's state diagram. However, given a collection of such CSM components, one can select a subsystem and obtain its product, representing (in one, large graph) all possible subsystem's executions or runs. Consequently, the model of a system can be subject to formal model checking methods and techniques. This advantage is not provided by a standard specification methods based on UML.

Moreover, as we have shown, the COSMA software environment supports the additional functional features, like stepwise model reduction, defining behavioral invariants, imposing time dependencies etc., as well as the means for the analysis of counterexamples. This makes the (Timed) Concurrent State Machines (and COSMA tool) a good candidate for a convenient framework for preliminary specification of concurrent, reactive systems. Once verified and corrected, such a specification can be refined, enhanced and otherwise developed in other professional software development environment. Moreover, if some components are to be *hardware*-implemented (which is often the case in embedded systems), the automata-like CSM specification is also close to common forms of behavioral specification of sequential circuits.

## References

1. Mieścinki J., Czejdo B., Daszczuk W. B. *Model checking in the COSMA environment as a support for the design of pipelined processing.* Proc. European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004, Jyväskylä, Finland, 24—28 July 2004.
2. Clarke E. M., Grumberg O., Peled D. A.. *Model Checking*, MIT Press, 2000.
3. Berard B. (ed.) et al. *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer Verlag, 2001.
4. Peled D. A.: *Software Reliability Methods*, Springer Verlag, 2001.
5. McMillan K. L.: *Symbolic Model Checking*, Kluwer Academic Publishers, 1993.
6. COSMA: www.ii.pw.edu.pl/cosma/
7. Gerth R., Kuiper R., Peled D. A., Penczek W.: *A Partial Order Approach to Branching Time Logic Model Checking*, Information and Computation, Vol. 150, No. 2 (1999), pp. 132-152.

8. Wolper P., Godefroid P.: *Partial-Order Methods for Temporal Verification*, in Proc. of CONCUR '93, Lecture Notes in Computer Science vol. 715, Springer-Verlag, New York, 1993, pp. 233-246.
9. Corbett J. C., Dwyer M.. B., Hatcliff J., Laubach S., Păsăreanu C. S., Robby Zheng H.: *Bandera: Extracting Finite-state Models from Java Source Code*, in Proc. of the 22nd International Conference on Software Engineering, ICSE 2000, June 4-11, Limerick, Ireland. ACM 2000, pp. 439-448.
10. Păsăreanu C. S., Dwyer M. B., Visser W.: *Finding Feasible Counter-examples when Model Checking Abstracted Java Programs,* in Proc. 7th International Conf. on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2001, Genova, Italy, April 2-6, 2001, Lecture Notes in Computer Science vol. 2031, p. 284.
11. Harman M., Danicic S. A: *New Approach to Program Slicing*, in Proc. of 7th International Software Quality Week, San Francisco, May 1994.
12. Mieścicki J.: *Multi-phase model checking in the COSMA environment.* Institute of Computer Science, WUT, Research Report 14/2003.
13. Mieścicki J., Czejdo B., Daszczuk W. B.:**.** *Multi-phase model checking in the COSMA environment as a support for the design of pipelined processing.* Institute of Computer Science, WUT, Research Report 16/2003.
14. Mieścicki J.: *Concurrent State Machines, the formal framework for model-checkable systems*, Institute of Computer Science, WUT, Research Report 5/2003.
15. Daszczuk W. B.: *Critical trees: counterexamples in model checking of CSM systems using CBS algorithm.* Institute of Computer Science, WUT, Research Report 8/2002, Warsaw, June 2002.
16. Daszczuk W. B., *Verification of temporal properties in concurrent systems*, Ph. D. Thesis, Faculty of Electronics and Information Technology, Warsaw University of Technology, January 2003.
17. Daszczuk W. B.: *Temporal model checking in the COSMA environment (the operation of TempoRG program),* Institute of Computer Science, WUT, Research Report 7/2003.
18. Alur R. Dill, D. L.: *A Theory of Timed Automata, in Theoretical Computer Science,* Vol. **126,** pp.183-235, 1994.
19. Alur R.: *Timed Automata. in 11th International Conference on Computer-Aided Verification,* LNCS **1633**, Springer-Verlag, 1999, pp. 8-22.
20. Daszczuk W. B.:. *Timed Concurrent State Machines,* Institute of Computer Science, WUT, Research Report 27/2003.

XXI Fall Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings, pp.205-212


# Automated Travel Planning

Piotr Nagrodkiewicz[1], Marcin Paprzycki[2]

[1] Warsaw University of Technology, Department of Mathematics and Information Science, Plac Politechniki 1, 00-661 Warsaw, Poland
nagroda100@wp.pl
[2] Computer Science Institute, SWPS, Chodakowska 19/31, 03-815 Warszawa, Poland
mpaprzycki@swps.edu.pl

**Abstract.** This paper summarizes the current state of art in the domain of automated travel planning. Requirements for planning systems are identified taking into account both functionality and personalization aspects of such systems. A new algorithm that allows planning routes between any two locations and utilizes various means of transportation is discussed.

## 1 Introduction

One of the areas served by information technology is the travel domain. Within this domain, much effort is being paid to the problem of integrating Internet available information. For instance, finding transportation between travel origin and destination is to be combined with locating suitable hotels, restaurants, etc. To achieve this goal information already available on the Internet must be integrated into a homogenous system that allows its exploitation. In this paper we address only one of aspects of developing an integrated *Travel Support System*– preparation of travel plans that combine multiple means of transportation. For the up-to-date description of a complete system, see [7].

Currently, within the Internet, there exist a number of systems that provide access to timetables and/or allow performing queries for routes between two given locations. However, in most cases such systems are limited to only one specific mean of transportation (airplane, train, bus etc.). A person who plans a trip involving an airplane and a train must manually search within multiple systems to find connections that together constitute a sensible travel plan. Within such a system proposed connection must satisfy user-defined and interrelation-based constraints. Separately, there exists software that supports traveling by combining GPS technology and digital maps. This software is capable of searching for the shortest routes and can help while traveling [19]. Unfortunately, this software is of value only for "individual" travels that utilize means of transportation such as cars, motorcycles etc. In other words, while it may be fun to watch on a GPS based display how the train is moving across France; such a system will not help us find connection with another train or a bus that we have to catch in Dijon. Obviously, combining travel from a "generic" origin to a "generic" destination (defined e.g. as a city name or a zip code) becomes more complicated when it involves particular targets like a detailed home address or a hotel. In this case the complete trip may involve one or more bus connections from home to the airport, flight to a specific destination, a train to the center of the city and a tram to the hotel. This is exactly the type of travel scenario that we are interested in.

In the paper we proceed as follows. In the next section we summarize information about most important travel planning systems. We follow with a description of our proposed approach (in Sections 3-7). In Section 8 we discuss some open research issues that have to be addressed for the proposed system to become truly robust.

# 2 Travel Planning

## 2.1 Current Research Projects

While there exists a number of travel planning projects [1, 2, 3, 5, 8, 9, 10, 11, 15, 16, 21, 23], two of them are of particular interest in the context of this paper. Both are a part of research conducted by Craig Knoblock and his collaborators at the University of Southern California. These projects are: *Heracles* and *Theseus* [21, 23]. *Heracles* is a framework designed to support creation of information agents that provide means of gathering and integration of data for a particular domain. An example of applying this framework (described at project's web pages) is a system called *Travel Assistant* that supports planning of business trips [11]. Provided with the origin and the destination of the travel, it is able to recommend best suiting means of transportation (taxi, private car or airplane). If the airplane is chosen, system advises whether it is cheaper to go to the airport by a private car and to pay for parking (for the time of being away on the trip) or to take the taxi. As an addition, *Travel Assistant* presents the weather forecast for the time of arriving at the destination and selects the closest hotel to the destination airport. *Travel Assistant* is able to optimize plan of travel against time or price. The *Theseus* project, on the other hand, allows mining information from sources available on the Internet–WWW pages in particular. It is also capable of monitoring over time the state of a task and it detects and reacts to changes. *Theseus* is based on flow charts, so it can execute queries in parallel and can also predict queries it will receive in the close future so it can start execution of them in advance increasing the speed of the whole system [2, 3]. Mining of information is achieved through wrappers dedicated to each source of information.

## 2.2 Hierarchical Task Networks

When speaking about planning it is difficult not to mention *Hierarchical Task Networks* [12, 13]. Here, the idea how to solve the travel problem is to divide getting from place A to place B into two or more tasks of getting from A to B through some intermediate locations. For instance, let us consider a person that wants to get from his or her home to work. This task might be decomposed into smaller subtasks: getting from home to the bus stop, getting from that bus stop to the bus stop next to the work and finally from the last bus stop to the place of work. This is a very simple example, however often a very complicated net of (sub)tasks could be obtained. Generally, decomposition is achieved by applying one of the predefined operators which involve conditions that specify precisely when they can be applied.

# 3 Planning Algorithm–the Idea

When planning a trip one must take into account a lot of information to achieve the desired goal. Since the space of possible solutions is extremely large, algorithm that is to be used to solve this problem has to proceed in such a way to maximally reduce the search space. We can start from an observation how people act when planning a trip. First, they try to find a 'general' route to the destination and once this "step" is completed they search for connections that satisfy the route that they have established. The proposed algorithm works in a similar manner. In the *first step*, it finds out divisions of the travel into stages (where each of them has a defined both the starting and end locations and a mean of transportation between these two locations.) In the *second step*, for each division algorithm searches for possible chains of real connections that implement it is performed. (Let us clarify, that when we talk about real connections we mean connections that involve real means of transportation, e.g. a train or a bus; however, currently the system Is not connected with any real-world travel resources; instead, a simulated world of travel has been created to test the proposed algorithm.) Resulting proposals are presented to the user. Algorithm is executed once and finds as many proposals as possible (at this stage no optimization is performed). Choosing the cheapest offer (or the fastest way) will be achieved by sorting all proposals against the desired criteria. Finding out all solutions will take more time then a dedicated search to optimize a single criterion, but will be generally faster then performing multiple searches to satisfy multiple requests that the user will very likely issue. Let us now describe the proposed algorithm in more detail.

# 4 Planning Algorithm–Step I

## 4.1 Conditions of Dividing the Trip into Stages

As described above, the task of the first step of the algorithm is to prepare a set of divisions (into stages) of the complete travel. We require that each stage must have defined (1) both start and end locations, and (2) mean of transportation used between these two locations. What is more, we require that all stages must end where the next stage in the division begins. For the first stage in the division we require that it starts as close as possible to the actual beginning of the travel and for the last stage in the division to end as close as possible to the actual end of travel (in both cases we call distances between these locations: *unplanned distances*).

## 4.2 Generation of Divisions

Let us now describe how divisions of a travel into stages are generated. Here, we utilize the idea of each travel having one main stage, for example, a flight or a travel by a train, and apply the *Hierarchical Task Networks* approach. We create a division by finding out the main part first, and solving two subtasks recursively (from start of the stage to the start of the main part; and from the end of main part to the end of the stage). By doing so, we eventually obtain a complete set of divisions. We construct more then one division, because there are many possibilities to choose the main stage (it is described in detail below). Recursive division ends at some level when the main part covers the whole distance being divided or when there is no possibility of further division. Having solved both subtasks it is possible to construct divisions for the given level–it is achieved by merging each of the divisions for the first subtask, the main stage and each of the divisions of the second subtask. Merging is possible only if (1) all stages satisfy conditions described before (geographic continuity of stages) and (2) the sum of unplanned distances on both ends of the result chain of stages is not greater then the distance between the start and end locations of the part being divided (since the division would increase the overall distance the plan is needed for).

## 4.3 Selecting Accepted Paths

When all divisions are ready it is possible to remove some of them. First of all, it may turn out, that some divisions are identical (this is a result of recursion and many possibilities at each level of it). Secondly, we remove all divisions that imply much longer distance then the rest of them. This is done by the means of statistics. Each division implies a distance at least as long as the sum of lines joining ends of each stage (since the straight lines are the shortest possible routes). Divisions with the implied distance, for example, two times longer then the average one might be removed as very often they are of lesser quality (they may be zigzagged). Furthermore, divisions that start and/or end not exactly where the travel does (*unplanned distances*) should be removed if there are divisions that start and/or end exactly where the travel does (ones without *unplanned distances*). We assume here that it is better for a user, if possible, to obtain only plans that start and end where he or she wanted to. This assumption may need to be relaxed though, if we take into account discount airlines that fly to somewhat remote places. If plans do not start or end exactly where the travel does (for example, if there are no stations known to the system in the area), it would be an improvement to remove these divisions that have *unplanned distances* much longer then the other ones, once again this is achieved by the means of statistics. Whenever removing possible paths it is assumed that the removal is performed only if some arbitrary number of them will be left (e. g. 10) so we never reduce the number of possibilities too far–a relatively wide variety of proposals is guaranteed to be ready for the processing in the second step of our planning algorithm.

## 4.4 Establishing Travel Details

Let us now describe how the main stages are found. At first, proper means of transportation are chosen using dedicated rules. These rules may decide that it is, for example, too long to go by walk, just enough to take a taxi or too close to fly (but in some other situation flying might be just fine, if there is an obstacle in the way that makes other possibilities impossible.) Generally, the rules decide which means of transportation would work in the given situation and they should tend to exclude the means of transportation that would not produce any possibilities in the second stage of the planning algorithm that is described later. Subsequently, for each of the

chosen means of transportation the start and end of the stage are set. If the mean of transportation has no stations (like walk or taxi) the stage covers the whole distance. If there are stations, at both ends we search for *N* stations laying the closest to the start (or end). Once they are identified, the main stages are created between every possible pair of start-end stations (under condition that such pair is not against rules used to choose proper means of transportation; before rules where applied to a different pair of locations—t may turn out that one of the chosen closest stations lays on another island, for example, and hence is not a wise choice any more.)

Rules applied to select specific mean of transportation may be based purely on distance between the two given locations–different means of transportation are better for different distances. However, if some extra information is added to the coordinates of those two locations, more sophisticated rules may be constructed. For instance, information on continent and an island/continental part on which the given location lays, may allow bringing better rules in–normally an airplane would not be used on short distances, unless there is an obstacle in the way (such as a sea). Such extended information may be gathered automatically for most stations and towns [20].

Use of rules causes the system to search for connections that lead in a rather direct way to the target location. This is generally a recommended behavior, however sometimes it might be undesirable. There might be only one complex solution that leads in a curve-like way to the target or there might be a very attractive price offer requiring such a route. Unfortunately, the planning system will not find it, as it will not even consider it. However, the presented algorithm allows receiving feedback that may be used to mediate such problems–it allows extending the list of stations searched for a given mean of transportation in the first step of algorithm by some additional pairs of stations. These new stations are marked to be coupled only with each other, not with any other station. Such additional pairs of stations allow making the algorithm to consider some cases it would have normally not investigated. In this way, for instance, we address the problem of discount airlines flying from Frankfurt Hahn rather than Frankfurt's main airport.

## 5 Planning Algorithm–Step II

In the first step of the planning algorithm divisions of travel into stages have been prepared. In the second step these divisions are being realized by real connections.

### 5.1 Implementation of Divisions

Stages of a single division are completed one by one. We start from the beginning of the trip and conditions at the end of each stage specify entry conditions for the next stage. For instance, arrival time at the station establishes set of possible trains. Of course, the first stage requires a special treatment, since there are no connections leading to it. Therefore some number of connections (e.g. 5) leaving from the start location is chosen as initial search connections (each with different next station in its schedule; and for each of them the earliest one is chosen). Then the search is performed until the connection arriving at the end location of the stage is found. Search is performed by simulating travel and checking possible changes at each visited station. Not all changes are checked as that would lead to browsing of far too many possibilities to perform the search in the acceptable time. Once again a set of dedicated rules is used to discard some of the potential changes to reduce the time spent on browsing of the search space. Rules are deciding whether a particular change should be ignored or not. They can do so taking into account various aspects: (1) These rules may easily discard connections that definitively do not lead in the right direction. (2) They may also enforce keeping of direct connections (so no unnecessary search is performed if a connection leading to the destination is found.) (3) Another rule may state that it is not wise to change to the connection that have the same route but is a later one, since this gains nothing. (4) It is also pointless to change to a local train if the current location is very far away from the target destination. Other rules may be devised as well. It is obvious that the more sophisticated rules will be, the faster the search will be performed (as they will eliminate a larger number of spurious connections).

We have to acknowledge that not all means of transportation do have stations and hence they have no timetables (e.g. taxi). As a result, when stage with such a mean of transportation is encountered we need to reserve a minimal required time for that stage, so that the start time for the next stage can be established. An exact duration of that stage will be known precisely when connections for the next stage will be found–it may turn out then, that there will be more time for the stage then previously reserved. It is crucial to always reserve ample amount of time for such stages, otherwise it may turn out that the whole plan will not be executable.

## 6 Planning Algorithm–Architecture

In Figure 1 we present architecture of the proposed planning system. Red arrows indicate a typical flow of information in the system: at first, for given locations, divisions of travel into stages are generated, subsequently these divisions are implemented with the real connections and eventually all the resulting proposals are being sorted according to expected user satisfaction (personalization).

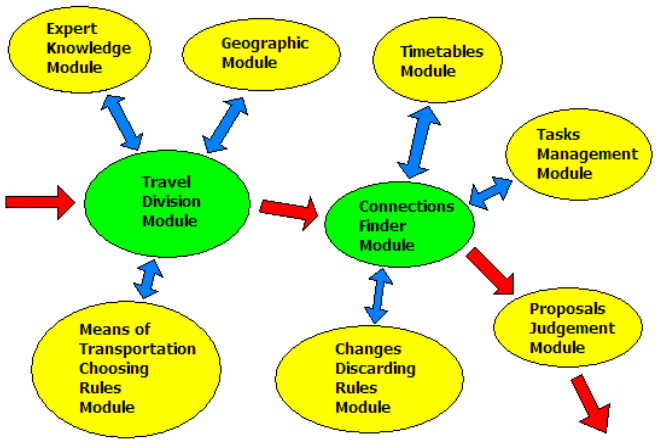



**Fig. 1.** Architecture of a travel planning system.

Each of system's modules might be encapsulated by a separate agent. These agents might be divided into groups that can operate on different computers, if there is such a need, for instance, for performance reasons. These groups are: agents working on travel divisions, agents finding connections and agents judging the prepared travel proposals. These groups should not be split as agents belonging to them exchange large amount of information (thus possibly generating a lot of network traffic).

### 6.1 Travel Division and Finding Connections

Divisions of travel into stages are generated with utilization of information provided by the following modules: geographic module (extended location information and finding of stations situated close to the given location), means of transportation selection module (a rule based system that provides list of appropriate means of transportation) and expert knowledge module (responsible for providing feedback to the first step of the algorithm, as described above). The test implementation of the planning system utilizes JESS (*Java Expert System Shell*) as the rule system [22]. This allows the dynamic edition of rules without making any changes to the remaining code of the system and thus experimenting with different sets of rules. Prepared divisions of travel into stages are passed to the connection finder module for realization using the real connections.

Realization of each proposed path is achieved with help of the following modules: timetables module (provides connections passing through the given station; it might also manage a database of such connections), tasks management module (decides in what order changes are investigated) and changes discarding rules module (a rule-based system deciding whether to ignore the given change, also JESS-based). Realizations of all paths constitute proposals of travel routes prepared for the user. They are passed to the judging module and sorted so that the best of them (according to system's knowledge of user preferences) are going to be presented first.

### 6.2 Proposals Judgment Module

This is the last of main modules of the planning system. Its task is to judge already prepared proposals and sort them based on likelihood of being attractive to the user. Judging proposals might be done by using case-based logic. Each travel proposal is described as a single case containing its most important characteristics [14, 18]– the more of them, the better. Cases, in turn, are stored in the *Case Retrieval Net* that allows fast retrieving cases similar to the given one. Furthermore, each case does not have to have the same descriptors as the other ones (e. g. one case can contain more visited stations while another case may contain details of a flight while yet another one may involved details f travel by a train); the net will still work perfectly. With each case there is

also associated a decision made by the user on the proposal it describes: it can be either acceptance (the user has chosen this proposal as his or her final plan) or the direct statement that a proposal is unacceptable (user has indicated that the given proposal is totally unacceptable to him or her). Each proposal being judged is transformed into a case, which in turn is compared against cases already stored in the net. Most similar case from the net is chosen. User may influence the process of computing of similarity between cases by defining the importance levels of each descriptor–this gives him or her opportunity to tune the mechanism, so it will take into account only criteria important to the user (with proper weights). Similarity of the most similar case to the judged one is taken as the judgment value (with a negative sign, if the decision associated with the most similar case was a rejection). In this way each of the proposals is judged and then all of them sorted based on that judgment. Subsequently, user browsing the list of proposals may reject some of them or choose one of them as a plan to be executed. In both cases an additional case is created and stored in the net along with the decisions made about it–this will extend the knowledge gathered in the net, hence over time the judging mechanism is able to learn preferences of the user–even such that would be very difficult to learn otherwise, like that in some regions user prefers trains and in other buses.

A screen-shot of the system as it has been implemented is presented in Fig. 2. Instead of a GIS module (which was not available) we have implemented our own "world editor" that was used to generate worlds characterized by features that the proposed algorithm had to deal with (e.g. islands, and peninsulas introduce special complexities to the system).

## 7 Planning Algorithm–Remaining Issues

*The optimal solution.* In both steps of the algorithm the rule-based systems are used to reduce the search space as far as possible. They prune the search tree considerably, however it may turn out that in some cases they prevent the algorithm from finding the most optimal solutions. This is an unavoidable trade off, since deciding not to use the rules would mean searching through the paramount number of possibilities what would in turn made the planning process cumbersome, if not prevent it from running at all in any sensible time.


**Fig. 2.** Screen-shot of the system as it is currently implemented

*Planning with date of departure and date of arrival*. As every trip planning system, algorithm presented here allows user to issue travel queries that may be either given date of departure or date on which he or she has to

reach the destination location. From the technical point of view, there are only small differences in the algorithm in both cases–only time definitions of 'next' and 'previous' stations are being swapped. In the case of planning to arrive at the given date, the planning takes place from the end toward the beginning–stages of travel divisions are being implemented from the last to the first.

*Parallel Computing*. It is worth mentioning that the proposed algorithm can be easily parallelized. Even though applied rules greatly reduce the search space, it still may turn out that there will be a massive number of possibilities to check. Hence this feature might be important in a production environment. Possibilities of parallel computation exist in both steps of algorithm. In the first one, subtask divisions at all levels might be found in parallel as these are independent divisions. In the second step each change might be investigated independently and thus in parallel.

## 8 Future Work

Let us now discuss directions that the existing algorithm (that has been implemented–see Fig. 2 for the screen-shot of its current interface–and is available from `http://mpaprzycki.swps.edu.pl/mp/cvr/research/agent.html`) can be extended.

*Planning trips between multiple destinations*. Presented algorithm allows planning of a route between any two random points. However, sometimes it might not be enough. For example, a person may be interested in visiting more the just two places. In this case, the proposed algorithm could be used to find routes between any two of these places. The problem is how to establish order of visiting these places (assuming that it is up to the system to make such a suggestion). It is not a trivial task as it is a case of the Traveling Salesman Problem (TSP). A simple solution is to ask the user in what particular order he or she wants to visit all locations. The other one is to use one of the approximation algorithms known for the TSP.

*Problems of Uncertainty in Time Reservations.* When describing the second step of the planning algorithm it was mentioned that there is a need to reserve some time when encountering stage(s) with no timetables (like a taxi) as it is not known how long these stages could last. While the idea to reserve some time for these stages is not so bad, the problem is that if the amount of time being reserved will turn out to be too short, the whole plan will become impossible to complete. On the other hand, reserving too much time would cause the plan to have unnecessary layovers. A simple solution to this problems is to utilize a distance measure and establish a "benchmark time" e.g. for a taxi use 15 minutes for 10 kilometers and then scale it linearly. Unfortunately, this approach has some serious disadvantages. First of all, how to establish the benchmark time (city or highway? day or night? etc.)? Second, when scaling, how to be sure that the real distance between the start and end of the stage is the same as the length of the straight line between these two locations? It might be possible that between these two locations there is a river and the bridge is in some distance. Finally, how to take into account the fact that every day on a given street there is a traffic jam between 4pm and 6:30 pm? Better solution to that problem must be devised.

*Planning without positions of stations*. To join different means of transportation there is a need to know if stations are close to each other (can we switch from the bus to the train easily? do we need to change trains stations–like in Paris or Vienna?). This is easy to determine only if geographic locations of stations are available. Without explicit joints between stations used in subsequent stages it is hardly possible to complete the plan. Assuming these joints could be somehow available, there still would be an unavoidable increase in the number of possibilities to check while searching, since only a smaller number of rules could be used. Of course, some simple geographic information could be implicitly gathered–for instance, the continent. But the name of island would still be a problem. Generally, geographic locations are crucial for the first step of the algorithm. They also influence the second step, but in theory it could work without them. Since geographic locations of all stations are not easily available at the moment, some research should be undertaken to try to devise some other means of finding out divisions of a travel into stages.

*Increasing user interaction.* Another interesting aspect of planning–connected to travel personalization–is increasing users' interactions with the system. This could be achieved either by letting them to modify the divisions prepared in the first step of the algorithm or by allowing modification of prepared proposals. Modifications of the first step of algorithm would allow user, for example, to have a route leading through his favorite city (possibly to meet with friends). From algorithm's point of view such modifications would be transparent. The only problem is a need to develop an intuitive graphical interface that would allow user performing such modifications. Modifications to plans generated in the second step of the algorithm could allow user to enforce use of some other connection, for example, if she decides the algorithm gave her not enough (or too much) time for some stages due to the time reservations or because she would like to have more time to have a stroll in the city. Modifications made to the prepared plans would, however, require re-planning of some parts of the route.

## 9 Concluding Remarks

In this paper current state of art concerning the automated planning of a travel has been described and some of their weaknesses have been specified. In response we have proposed an algorithm that allows preparing travel plans utilizing multiple means of transportation. This algorithm is not limited by distances between locations it is to plan the route for. Architecture of the solution allows easy modification and replacing modules responsible for different aspects of travel planning. Use of *Java Expert System Shell* allows modification and conducting experiments with any set of rules used to choose appropriate means of transportation when dividing the travel into stages, and to discard some solutions proposed when realizing these divisions with real connections. The proposed algorithm has been implemented and readers are invited to download it and experiment with. In the near future we plan to utilize it within the context of a travel support system described in [7].

## References

1. Strahan R.; Muldoon C.; O'Hare G. M. P.; Bertolotto M.; Collier R. W.: *An Agent-Based Architecture for Wireless Bus Travel Assistants,* http://emc2.ucd.ie/publications/WIS03.pdf [accessed: 15 October 2004]
2. Barish Greg; Knoblock Craig A., *Learning value predictors for the speculative execution of information gathering plans,* http://www.isi.edu/info-agents/papers/barish03-ijcai.pdf [accessed: 10 September 2004]:
3. Barish Greg; Knoblock Craig A., *Speculative execution for information gathering plans* [accessed: 10 September 2004] http://www.isi.edu/info-agents/papers/barish02-aips.pdf
4. Cheyer Adam; Julia Luc, *Multimodal Maps: An Agent-Based Approach* [accessed: 23 September 2004] http://www.adam.cheyer.com/papers/lnai1374.pdf
5. Dillenburg John F.; Wolfson Ouri; Nelson Peter C.: *The Intelligent Travel Assistant,* [accessed: 12 September 2004] http://www.cs.uic.edu/~wolfson/mobile_ps/ita02.pdf
6. Ambite Jose Luis; Barish Greg; Knoblock Craig A.; Muslea Maria; Oh Jean; Minton Steven: *Getting from here to there: Interactive planning and agent execution for optimizing travel,*. [accessed: 10 September 2004] Available in World Wide http://www.isi.edu/info-agents/papers/ambite02-iaai.pdf
7. Gordon M.; Paprzycki M., *Designing Agent Based Travel Support System* [accessed: 25 September 2005 http://www.cs.okstate.edu/~marcin/mp/cvr/research/ISPDC_2005.pdf
8. Dillenburg John F.; Wolfson Ouri; Nelson Peter C.: *The Intelligent Travel Assistant* [accessed: 12 September 2004] http://www.cs.uic.edu/~wolfson/mobile_ps/ita02.pdf
9. Kay Michael G.; Jain Ashish: *Issues in Agent-based Coordination of Public Logistics Networks,* [accessed: 24 September 2004] http://www.ie.ncsu.edu/kay/pln/IETR02-01.pdf
10. Kumar Praveen; Reddy Dhanunjaya; Singh Varun: *Intelligent transport system using GIS* [accessed: 25 September 2004] http://www.gisdevelopment.net/application/Utility/transport/pdf/164.pdf
11. Knoblock Craig A.; Minton Steven ; Ambite Jose Luis; Muslea Maria; Oh Jean; Frank Martin: *Mixed-initiative, multi-source information assistants*, http://www.isi.edu/info-agents/papers/knoblock01-www.pdf [accessed: 10 September 2004]
12. Nau Dana S., *Automated Planning: Theory and Practice*, chapter 11: Hierarchical Task Network Planning [accessed: 1 November 2004] http://www.cs.umd.edu/~nau/cmsc722/notes-fall-2004/chapter11.pdf
13. Nau Dana S., *Ordered Task Decomposition: Theory and Practice [accessed: 1 November 2004]* http://prometeo.ing.unibs.it/sschool/slides/nau/nau1.ppt
14. Peuret Frederic: *Case-Based Travel Agent,* [access: 28 August 2004] http://www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-69.pdf
15. Stallard David, *Talk'n'Travel: A Conversational System for Air Travel Planning,* [acceseds: 29 August 2004] http://acl.ldc.upenn.edu/A/A00/A00-1010.pdf
16. Homb Andrew; Mundhe Manisha; Kimsen Sonali; Sen Sandip: *Trip-planner: An Agent Framework for Collaborative Trip Planning,* http://www.cs.wright.edu/people/faculty/mcox/mii/papers/manisha.pdf [accessed: 1 September 2004] .
17. Tuchinda Rattapoom; Knoblock Craig A.: *Agent Wizard: Building Information Agents by Answering Questions,*. [accessed: 10 September 2004] http://www.isi.edu/info-agents/papers/tuchinda04-iui.pdf
18. Waszkiewicz Paweł; Cunningham Padraig; Byrne Ciara, *Case-based User Profiling in a Personal Travel Assistant,* [accessed: 29 August 2004] http://www.cs.usask.ca/UM99/Proc/short/WaszkiewiczP.pdf
19. *Automapa* http://www.automapa.com.pl/ [accessed: 20 July 2005]
20. *Getty Thesaurus of Geographic Names* [acceseds: 5 February 2005] http://www.getty.edu/research/conducting_research/vocabularies/tgn/
21. *Heracles: Constrain-based Integration* http://www.isi.edu/info-agents/Heracles/ [accessed: 10 September 2004]
22. *Java Expert System Shell* http://herzberg.ca.sandia.gov/jess/ [access: 5 September 2005]
23. *Theseus: Plan Execution* http://www.isi.edu/info-agents/Theseus/ [access: 10 September 2004].

XXI Fall Meeting of Polish Information Processing Society
Conference Proceedings, pp.213-221
ISBN 83-922646-0-6


# KTDA: Emerging Patterns Based Data Analysis System

Roman Podraza[1], Krzysztof Tomaszewski[2]

Warsaw University of Technology, Institute of Computer Science
15/19 Nowowiejska, 00-665 Warsaw, Poland
[1] R.Podraza@ii.pw.edu.pl, [2] k_tomaszewski@users.sourceforge.net

**Abstract.** Emerging patterns are kind of relationships discovered in databases containing a decision attribute. They represent contrast characteristics of individual decision classes. This form of knowledge can be useful for experts and has been successfully employed in a field of classification. In this paper we present the KTDA system. It enables discovering emerging patterns and applies them to classification purposes. The system has capabilities of identifying improper data by making use of data credibility analysis, a new approach to assessment data typicality.

## 1 Introduction

Knowledge discovery or data based inference is one of the most important purpose of accumulating data and maintaining large, often only growing, databases. Emerging patterns (EPs) [2] are examples of special relationships observed on attribute values of items. EPs can be then analyzed by experts (supported by computer systems) to discover new rules or relations in a given domain to understand it better. For instance EPs can be exploited for classification purposes. It seems nowadays almost no one has to be convinced of benefits of data mining and knowledge discovery, especially in the business world.

But all data analysis and knowledge discovery make sense only if processed data are credible. At first, to ensure the most possible data credibility, validity and consistency checks are used at data gathering stage. Then in most cases, a large number of processed records are analyzed to gain some generalized information, facts, rules. There is an unspoken assumption that most of data are correct, thus a minor, not credible part of considered dataset will not disrupt discovered knowledge too much. Often there is so much of data that we can reject some of them by applying some data cleaning procedures without much information loss. However, there exist still some applications where such approach is inappropriate. As an example one can point medicine [1], where a single record of a database can often represent an individual patient. In such a case no records can be removed, even if there are indications that data may be corrupted. Moreover in such sensitive domains data credibility gets its special significance. If the data based inference can have any influence on medical decisions, it is obvious that a particular care must be taken to ensure or at least asses data credibility. One of possible approaches is to employ some data credibility estimation mechanism which will pay expert's attention to records, which seem to be most incredible.

In this paper we present the KTDA (shortening for *KT Data Analysis*) system. It is user-friendly tool for discovering emerging patterns in data. The KTDA system implements two different algorithms of discovering emerging patterns, proposed in [2] and [3], but with some extensions and improvements. EPs enable data classification for which CAEP algorithm [4] is applied. Moreover, with KTDA system it is possible to assess data credibility using credibility coefficient, as proposed in [5] and [6]. In the KTDA system an original credibility coefficient computing algorithm was implemented. It takes into account data characteristics expressed by discovered emerging patterns. Its details are going to be published elsewhere. The paper is organized as follows. In Section 2 a short description of emerging patterns is given. Then, in Section 3, a brief introduction into data credibility analysis is submitted. After presenting in Section 4 an overall view of the KTDA system and its capabilities the paper is completed with some conclusions.

## 2 Emerging Patterns

Emerging patterns are closely related to frequent patterns, widely known as *frequent itemsets* [7]. Both are kinds of relations on attribute values discovered in datasets and both have the same form. In this paper we define a dataset as a set of data records, each described with the same set of attributes which can be continuous (numeric) or nominal (discrete).

A pattern consists of some terms which, in fact, are individual conditions or, in other terminology, true-false tests. Each condition refers to a single dataset attribute and determines a set of values of this attribute satisfying this condition. In most cases conditions for continuous attributes check whether attribute's value is less-equal or greater then the given thresholds. A condition for a nominal attribute checks if its value is equal or not equal to a certain constant. A particular record agrees with the whole pattern if and only if it satisfies all conditions contained in this pattern. Then we say the pattern matches to this record. The ratio of the number of records matched by the pattern to the number of all records in the considered dataset is named the pattern *support*.

If we are interested what attribute's values often appear jointly we would like to discover in our dataset some patterns with a support high enough (greater or equal then a specified support threshold). These are frequent itemsets and they describe some characteristic features, states or relations in the dataset (at the given support threshold). Now let us assume that our dataset contains a decision attribute. This is a typical nominal attribute but its value denotes association of the given record to a group of records with the same value. These disjoint groups of records create decision classes. For example, *diagnosis* can be a good decision attribute dividing some patients' dataset into two decision classes: *healthy* ones and *ill* ones. Now if we are interested what is distinguishing in one of these decision classes the frequent itemsets are not sufficient. Some of these patterns could be common to both decision classes (high support in both classes) and do not represent a knowledge describing only *healthy* class or only *ill* class. Really interesting are these patterns which have a high support in one decision class and at the same time a low support in the other one. To distinguish two decision classes it is desirable to find out such patterns which are frequent itemsets in one class and are infrequent in the other one. These patterns are just called emerging patterns. The decision class in which an EP has a higher support is referred to as a *target class* for this EP. In more general situation there are *N* decision classes and we are interested in discovering EPs for each decision class as their target class. In this case for each decision class we are composing a temporary division of the dataset into two subsets, a first one consisting only of records belonging to the decision class and a second one consisting of all other records (the rest of the dataset). The ratio of the pattern support in its target class to the pattern support in the rest of the dataset is a *growth rate* for this pattern.

How high should be EP's support in its target class and how low in the rest of the dataset? Actual values of support are not important. Essential is the EP's growth rate. Larger values of the growth rate denote more characteristic EPs for its target class. In the approach proposed in [2] a growth rate threshold (greater then 1) is arbitrary chosen and only these EPs which have the growth rates greater or equal to that threshold are discovered. As a result we can obtain many EPs with quite low values of both supports and still having satisfactory value of the grow rates.

The other approach [3] is to detect only EPs with sufficient statistical significance. In this methodology the growth rate threshold is of no importance and a significance level value parameterizes the set of results (EPs). The significance level value is used then in a process of statistical hypothesis testing to assess statistical significance of each inferred EP and not significant EPs are rejected. In consequence we can acquire many EPs with lower growth rates but with higher supports and we have got the guarantee that they are all statistically significant at the specified level.

These two approaches lead to different sets of EPs generated from the same dataset although obviously many of patterns are the same or similar. In the KTDA system the both methods of discovering Emerging Patterns have been implemented. The first of them utilizes maximal frequent itemsets approach [7]. Details of the algorithm can be found in [2]. The second method makes use of decision trees [8]. The exact Fisher's test [9] is employed in procedure of decision tree construction as a statistical test for assessing significance. This algorithm was proposed in [3].

## 3 Data Credibility Analysis

Data credibility analysis is a new research area in a domain of knowledge acquisition. The main goal of the research is estimating credibility of individual records of analyzed datasets and applying this expertise for ensuring maximal data credibility. Evaluation of credibility of data is done by specialized heuristic algorithms. Some of them were described in [5] and [6]. The most important aspect of these algorithms is unawareness on meaning of the processed data. This makes them general, universal and ready to operate on any data. Basing on a given dataset they assign to each data record the relative credibility estimation known as a credibility coefficient.

This is just a real number in range [0, 1]. Lower values indicate lower estimated credibility. The intention of the proposed data credibility assessment algorithms is to assign lower credibility coefficients to less typical record. They are commonly invalid, outlying or abnormal data. In any of these cases it is good to identify such records. Invalid data are obviously incredible and outlying data do not match well to typical schemes so cannot be used to infer a general knowledge. For example if in a medical application an outlying patient record denotes a special case, he or she is going probably to be treated with some extra care and most likely will get slightly different remedies. Since calculated credibility coefficients are relative to the analyzed dataset the system itself cannot decide how low coefficient value denotes an incredible record. Nevertheless an expert can revise a chosen number of records (for example: 10% of the dataset) which were given the lowest credibility coefficient. Then he/she can make the decision how significant are the records and what to do with them (e. g. neglect, correct, start thorough investigation of cases).

The KTDA system contains our two new, general algorithms of computing credibility coefficients: *the Voting Classifier Method* and *the Multi Credibility Coefficient Method*. They are general because their parameters are other algorithms. They will be described in details elsewhere.

The Voting Classifier Method computes credibility coefficients by using a voting classifier. In the KTDA system it uses the CAEP (Classification by Aggregating Emerging Patterns) classifier [4], which is a voting one. In this way EPs can be exploited in data credibility analysis. Some other kinds of voting classifiers, such as neural network, SVM, k-NN, Bayesian classifiers, etc., are planned to be exploited as well for the Voting Classifier Method.

The Multi Credibility Coefficient Method allows obtaining credibility coefficients as an aggregation of many credibility coefficients computed by an arbitrary number of algorithms. The main idea of proposing this solution was to gain all advantages of various approaches. Different credibility coefficient computing algorithms produce better results in different cases. Usually it is impossible to choose the best one among them. Instead of choosing one such algorithm it would be better to use them all and benefits from their individual advantages. This is exactly what the Multi Credibility Coefficient Method is performing. Our initial experiments have shown that this approach allows obtaining even better results then the best outcome of a single method, which is incorporated into the Multi Credibility Coefficient Method. In current version of the KTDA system the Multi Credibility Coefficient Method has a fixed configuration consisting of two Voting Classifier Methods basing on CAEP and differing in algorithms they use to discover EPs.

## 4 System Overview

The KTDA system has been developed for 1.5 year. It has a comfortable graphic user interface and its source code level portability (C++) enables implementations under many different operating systems. The KTDA system has been successfully used under Linux and MS Windows.

The KTDA system has multi-window interface architecture but its main window plays a key role in controlling the execution of the program and managing other information windows. The KTDA system's main window is shown in Fig. 1 and Fig. 2. It has a very simple and intuitive interface consisting of the main menu and two views: *Object* and *Windows*. In most cases only the *File* menu from the main menu is used for opening and closing datasets. The other functions in the main menu of the KTDA system covers experiments associated with data credibility analysis. There are as well some options which do not affect KTDA results anyhow. The *Windows* view plays only supporting role and allows bringing up and down or closing other KTDA windows. Thus the most important element of the main window is the *Objects* view. It shows a hierarchical view of all objects created and processed during applying the KTDA system: opened dataset, defined decision systems (dataset with set decision attribute), discovered emerging patterns, computed credibility coefficients, CAEP classifiers and classification results. Each of these object types have their own icons so the view is very comprehensible. All operations the user can accomplish on a given object are accessible from a context menu appearing after clicking the right button of the mouse while pointing on the object. For most object types the context menu contains the *View* and the *Properties* items. Choosing the *View* item the user opens a new, dedicated view window presenting information characteristic for the selected object. Depending on the type of the selected object view window provides an additional context-dependent functions. Example view windows are shown in Fig. 3, Fig. 4 and Fig. 5. Choosing *Properties* item from the context menu the user get access to some detailed information on the given object.

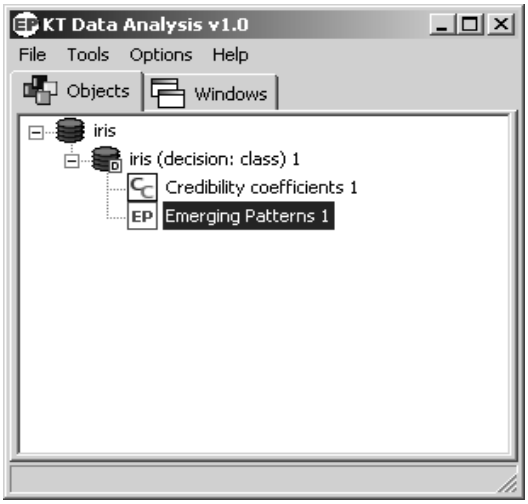



**Fig. 1.** Main window of the KTDA system under MS Windows. Object view contains: dataset object (*iris*), decision system object (*iris (decision: class) 1*), credibility coefficients object (*Credibility coefficients 1*) and Emerging Patterns object (*Emerging Patterns 1*)

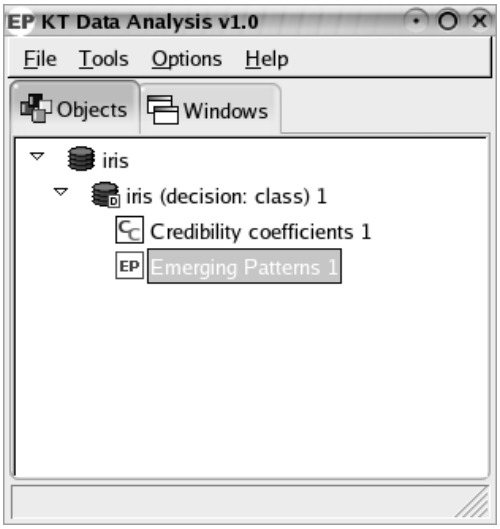



**Fig. 2.** Main window of the KTDA system under Fedora Core (Linux) with GNOME window manager. Object view contents as described for Fig. 1

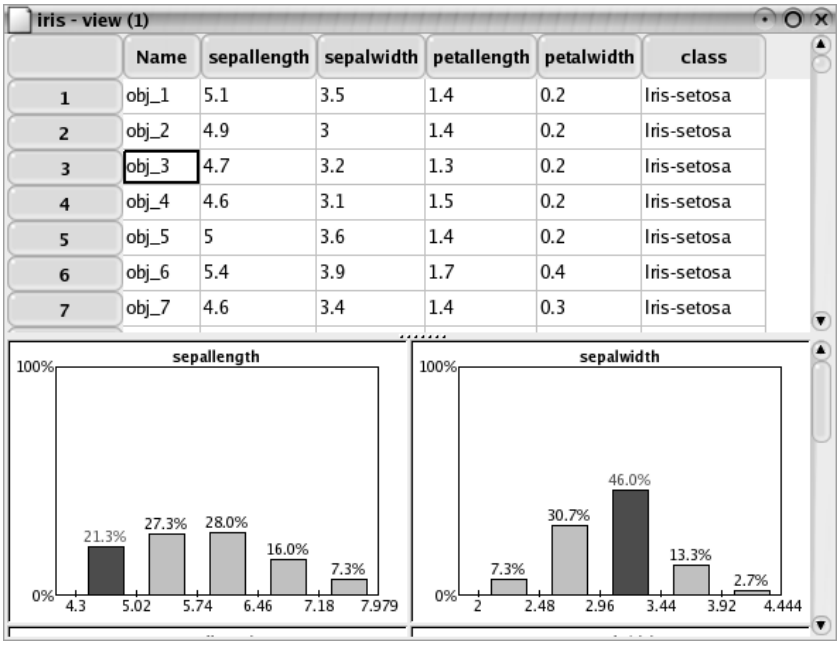



**Fig. 3.** Dataset view window. Bottom part of the window contains histograms showing value distribution of individual attributes. Histogram bins related to a record chosen in the table are marked.

### 4.1 Loading a Dataset

To start working with the KTDA system one has to decide on a dataset to be analyzed. There are two possibilities: a dataset can be open from a file or generated by the KTDA system itself (the KTDA system supports two types of synthetic datasets). The second case is related mainly to performing comparative experiments, with artificial datasets having the required and known characteristics. The KTDA system can be used for example as a generator of datasets with multivariate Gaussian distribution. KTDA can read data files in following formats: ARFF (*WEKA* program files) [11], CSV (compliant with spreadsheets like MS Excel), DATA (UCI Repository) [12] and TAB (*RSES 2* program files) [13]. This allows comparative studies with other classification results of many other systems as well as processing of already existing datasets.

Finally the user must choose a decision attribute which will divide the loaded dataset into decision classes. The operation is commenced by choosing *Create a decision system* option from the dataset context menu. In the KTDA system one can define many decision systems with different decision attributes what allows data analysis from many perspectives.

### 4.2 Discovering Emerging Patterns

Discovering EPs is available through *Discover Emerging Patterns By...* item from the decision system object context menu. There are two algorithms to choose: *Maximal frequent itemsets based algorithm* and *Decision tree based algorithm*. Selecting one brings up a particular configuration dialog. The algorithm based on maximal frequent itemsets requires four parameters:

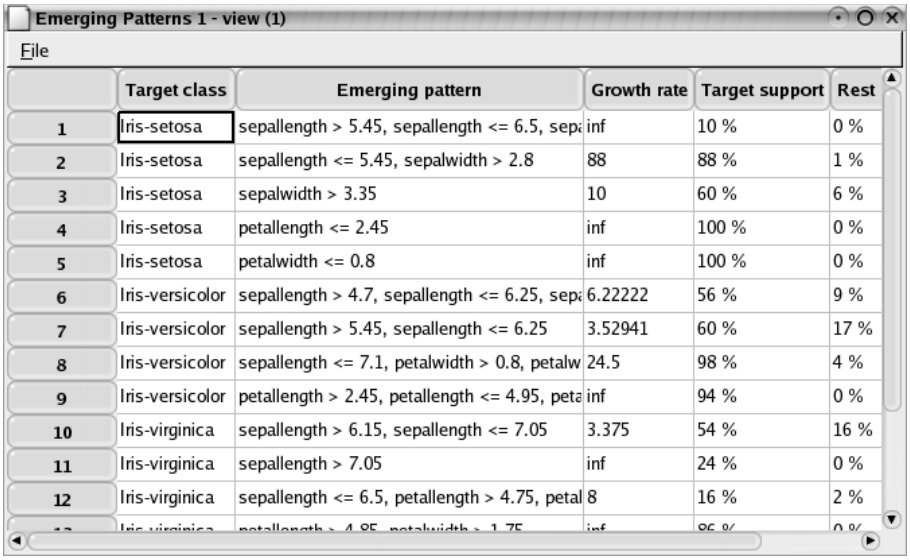
- *Minimal EP Growth Rate* – growth rate threshold for mined EPs;
- *Minimal EP support in target class* – specifies an initial support threshold in EPs' target classes;
- *Minimal-EP-support increase per iteration* – specifies a support threshold increase per main algorithm iteration. Each iteration runs with EP support threshold in target class calculated as support threshold from previous iteration (starting with value equal to *Minimal EP support in target class*) increased by this parameter value. Less this parameter is, more iterations are performed and more EPs can be discovered;
- *Reduce discovered EPs* – a two-stage switch whether to reduce set of discovered EPs or not.

Default values of these parameters should give the best results with relatively short computing time for most cases. All parameters in the KTDA system can be set through comfortable and easy to use dialog windows.

EP discovering algorithm based on a decision tree has a much simpler parameterization. Moreover, our experiments have shown that this algorithm is significantly insensitive to values of the parameters, so the default ones should be sufficient almost in every case. These parameters are as follows:

- *Split significance level*–determines a significance level used in checking the significance of splits considered during a decision trees constructing;
- *EP significance level*–a significance level used to test if EPs extracted from decision trees are statistically significant.

The user can examine discovered EPs with their growth rates and supports in target classes and in the rest of the dataset. EPs view window is shown in Fig. 4. The KTDA system allows exporting them to a CSV file (through menu *File* in the view window). By choosing *Create a CAEP* option in the context menu of EPs object one can obtain a CAEP object. It may be used to conduct a classification of dataset objects. It may be also used to compute credibility coefficients through the Voting Classifier Method. But the KTDA system provides much shorter and more practical way to do this. It is described in the next section.




| | Target class | Emerging pattern | Growth rate | Target support | Rest |
|---|---|---|---|---|---|
| 1 | Iris-setosa | sepallength > 5.45, sepallength <= 6.5, sepa | inf | 10 % | 0 % |
| 2 | Iris-setosa | sepallength <= 5.45, sepalwidth > 2.8 | 88 | 88 % | 1 % |
| 3 | Iris-setosa | sepalwidth > 3.35 | 10 | 60 % | 6 % |
| 4 | Iris-setosa | petallength <= 2.45 | inf | 100 % | 0 % |
| 5 | Iris-setosa | petalwidth <= 0.8 | inf | 100 % | 0 % |
| 6 | Iris-versicolor | sepallength > 4.7, sepallength <= 6.25, sepa | 6.22222 | 56 % | 9 % |
| 7 | Iris-versicolor | sepallength > 5.45, sepallength <= 6.25 | 3.52941 | 60 % | 17 % |
| 8 | Iris-versicolor | sepallength <= 7.1, petalwidth > 0.8, petalw | 24.5 | 98 % | 4 % |
| 9 | Iris-versicolor | petallength > 2.45, petallength <= 4.95, peta | inf | 94 % | 0 % |
| 10 | Iris-virginica | sepallength > 6.15, sepallength <= 7.05 | 3.375 | 54 % | 16 % |
| 11 | Iris-virginica | sepallength > 7.05 | inf | 24 % | 0 % |
| 12 | Iris-virginica | sepallength <= 6.5, petallength > 4.75, petal | 8 | 16 % | 2 % |

**Fig. 4.** View window for discovered Emerging Patterns

### 4.3 Computing Credibility Coefficients

To calculate the credibility coefficients one simply chooses *Compute credibility coefficients* item from a decision system context menu. Then there are two choices of algorithm to be used: *Voting classifier method based on CAEP classifier* or *Mutli credibility coefficient method*. In the first method there is one more dialog consisting of choosing EP discovering algorithm and configuration parameters of this algorithm. It was described in the previous section. Since in the current implementation of the KTDA system *Mutli credibility coefficient method* has a fixed configuration, in the second case there is nothing more to set up. After computations a new credibility coefficients object appears in the *Objects* view of the main window. The *View* menu item from the credibility coefficients object context menu launches a specialized view window (Fig. 5). Marking of records with the lowest values of credibility coefficients attracts attention of the user to the data requiring a special care and/or handling. There are two modes of record marking. The user can select marking of all records that have credibility coefficient values less or equal to a given threshold. The second option is marking a specified part of the dataset, consisting of records with lowest credibility coefficients. Especially the latter mode seems to be useful as we rather would like to inspect some minor fraction of all records that are probably the most incredible.

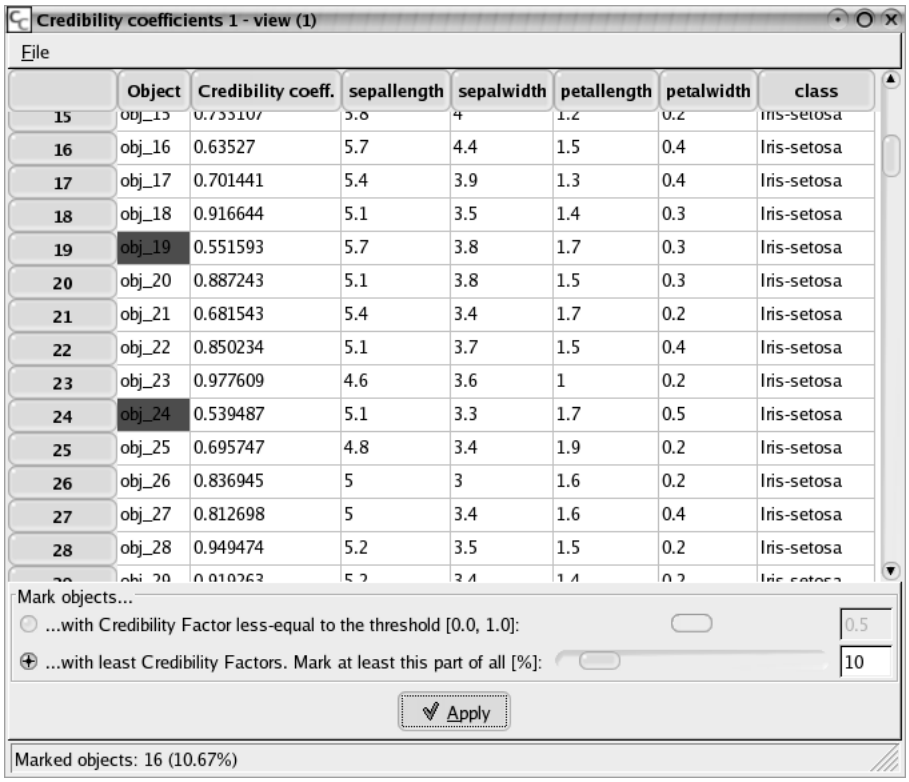



**Fig. 5.** Credibility coefficients view window. Among visible ones records *19* and *24* are marked according to marking condition specified in the bottom panel of the window

As it was described above the whole process of data credibility analysis using the KTDA system is simple and easy and does not demand any sophisticated knowledge from the user. Even quite inexperienced user can process data with the KTDA system to recognize the records seeming to be not typical and having the lowest estimated credibility. The decision what to do with such records is up to the user.

### 4.4 Other Functions

The KTDA system has some more auxiliary functions. These aid in performing many experiments with discovering EPs, classification based on EPs and credibility coefficient calculation algorithms. They have been used in different research undertakings. For example these auxiliary capabilities maintain adding some false, randomly generated records to the loaded dataset and checking whether they were properly identified by relatively low credibility coefficient values.

The KTDA system can be used as a data analysis system or as a research and educational tool. It supports performing the following automatic experiments:

- *classification experiment*–it can be carried out to observe how modifications of a given parameter of a particular EPs discovering algorithm influence the quality of classification accomplished by the CAEP classifier on a basis of the revealed patterns;
- *false object detection experiment*–it's purpose is to test how many generated false records are successfully identified by a certain credibility analysis method in respect to a number of false records inserted to a genuine dataset and parameters for the false record generator;
- *credibility coefficient and probability experiment*–it is performed to analyze correlations between credibility coefficient values and probability values for records of generated synthetic datasets, in which the probabilities are known. Such experiments are carried out to prove and/or assess a correctness of algorithms for evaluation the credibility coefficients - lower credibility coefficients should be assigned to less probable (more unusual) records.

These are *automatic* experiments, since each of them can be automatically repeated a required number of times and the results from all iterations are averaged to circumvent an influence of random fluctuations caused by applying a pseudorandom number generator. Other (non-automatic) experiments require some planning and user assistance.

### 4.5 Technology

The entire KTDA system was implemented in ISO C++ programming language [14] what benefited in high performance and source code level portability. The portability is preserved even by the graphic user interface as it utilizes *wxWidgets* [10] library–a portable and open-source GUI toolkit. The system can be compiled on almost any platform that has a contemporary C++ compiler and the standard C/C++ library. Implementations of the KTDA system were run under MS Windows and Linux (Fedora Core 3) operating systems.

All tools and libraries needed to compile KTDA are free and open-source. By choosing Linux operating system and GCC compiler one obtains absolutely free and stable platform for using the KTDA system. Moreover the KTDA system has relatively low hardware requirements. For quite a long time it has been developed on a machine with only 64 MB of RAM and a CPU of 400 MHz.

## 5 Conclusions

The KTDA system general description and its capabilities were presented in the paper. The KTDA system is technologically advanced but easy to use and user-friendly tool for data analysis. Its fundamentals are based on emerging patterns concept, relatively novel form of knowledge discovered in databases. Some introductory information on emerging patterns was submitted in Section 2. Two different algorithms for discovering emerging patterns were put into practice in the KTDA system. Comparative studies of these two approaches can be very beneficial for researchers and experts.

The KTDA system is also a tool for data credibility analysis. The paper presents essentials of the research and briefly explains its target and a methodology of credibility coefficients. The KTDA system suppors our two innovative algorithms for computing credibility coefficients: *Voting Classifier Method* and *Multi Credibility Coefficient Method*. The first one of them employs emerging patterns in generating the measure of credibility. The second algorithm is much more general and applies cooperation of many credibility coefficient calculating methods to obtain better results of credibility coefficients. The KTDA system only partially utilizes its advantages as in a current version it support only *Voting Classifier Method* with different parameterizations (EP discovering algorithm). We believe that *Multi Credibility Coefficient Method* used with broader set of credibility coefficients computing algorithms will increase data credibility analysis quality.

The system can be employed to work with almost all data having a tabular form, for example stored in a CSV file. A presence of predefined decision attribute is not required as the program allows to temporarily define one. The ability to define many different decision attributes enables to discover emerging patterns related to different aspects of processed data. Although the medicine was the primary inspiration for a data credibility analysis research the KTDA system is suitable not only for medical applications. It is universal and can be applied in almost every domain.

To fairly evaluate advantages and drawbacks of the KTDA system some more experience has to be gained. The perspectives are promising. The KTDA system is an interesting novelty in the field of data classification. The rules inferred from the dataset can be supplemented by the exceptions identified by credibility assessment tools. Experiment-oriented bias of the KDTA system makes it attractive for research and educational purposes.

## References

1. Roman Podraza, Piotr Ryszkowski, Wojciech Podraza: *Ignoring improper data in decision support system for medical applications*, Annales UMCS Informatica AI 2 (2004) 163-172.
2. Guozhu Dong, Jinyan Li: *Efficient Mining of Emerging Patterns: Discovering Trends and Differences*, Proceedings of the SIGKDD (5th ACM International Conference on Knowledge Discovery and Data Mining), San Diego, USA (1999) 43-52.
3. Anne-Laure Boulesteix, Gerhard Tutz, Korbinian Strimmer: *A CART-based approach to discover emerging patterns in microarray data*, Bioinformatics Vol. **19** no. 18, Oxford University Press (2003) 2465-2472.
4. Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, Jinyan Li: *CAEP: Classification by Aggregating Emerging Patterns*, Proceedings of 2nd International Conference on Discovery Science, Tokyo, Japan (1999) 30-42.
5. Roman Podraza, Adam Jurkowski: *Coefficient of Credibility in Rough Set System*, The 22nd IASTED International Conference on Artificial Intelligence and Applications (AIA), Innsbruck, Austria (2004) 776-781.
6. Roman Podraza, Mariusz Walkiewicz, Andrzej Dominik: *Credibility Coefficients in ARES Rough Set Exploration System*, *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing* 10th International Conference (RSFDGrC), Regina, Canada (2005) 29-38.
7. Karam Gouda, Mohammed J. Zaki: *Efficiently Mining Maximal Frequent Itemsets*, ICDM, San Jose (2001) 163-170.

8. John Shafer, Rakesh Agrawal, Manish Mehta: *SPRINT: A Scalable Classifier for Data Mining*, Proceedings of the 22nd VLDB Conference, Bombay, India (1996) 544-555.
9. Eric W. Weisstein: *MathWorld–A Wolfram Web Resource.* CRC Press LLC, 1999. Wolfram Research Inc. 1999-2004. http://mathworld.wolfram.com (2004).
10. Julian Smart, Robert Roebling, Vadim Zeitlin, Robin Dunn, et al: *wxWidgets–a portable C++ GUI toolkit* (2005) http::/www.wxwidgets.org
11. *WEKA – Waikato Environment for Knowledge Analysis*, Department of Computer Science, University of Waikato, New Zeland (2004) http://www.cs.waikato.ac.nz/ml/weka
12. Blake, C. L. & Merz, C. J.: *UCI Repository of machine learning databases,* Irvine, University of California, Department of Information and Computer Science (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html
13. *RSES2–Rough Set Exploration System,* Institute of Mathematics, Warsaw University (2004) http://logic.mimuw.edu.pl/~rses
14. Bjarne Stroustrup: *The C++ Programming Language,* Third Edition, Addison-Wesley (1997).

XXI Fall Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 223–231 

# Polymorphism—Prose of Java Programmers

Zdzisław Spławski

Institute of Applied Informatics,
Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland.
zdzislaw.splawski@pwr.wroc.pl

**Abstract.** In Java programming language as implemented in JDK 5.0 there appear rather advanced kinds of polymorphism, even if they are hidden under different names. The notion of polymorphism unifies many concepts present in typed programming languages, not necessary object-oriented. We briefly define some varieties of polymorphism and trace them in Java. Java shows that "industrial" programming languages are able to express more abstract patterns using rather involved theoretical means, hence the "working programmer" has to be better educated in order to understand them, recognize them in different programming languages under different names and superficial syntax, and make good use of them.

*Monsieur Jourdain.*
*Par ma foi! il y a plus de quarante ans que je dis de la prose sans que j'en susse rien, et je vous suis le plus obligé du monde de m'avoir appris cela.*
*Molière*

## 1 Introduction

Polymorphism ⟨gr. $\pi o \lambda \upsilon \varsigma$ = many + $\mu o \rho \varphi \eta$ = form⟩ in general means "multiform" and allows the same code to be assigned multiple types. This may be achieved in many ways, hence there are varieties of polymorphism in computer science and still new kinds of it are being proposed, but most of them are known only to theoreticians. The unqualified term "polymorphism" may cause some confusion, since among programmers it is often used to mean concrete kinds of polymorphism. For object-oriented programmers it almost always means inclusion polymorphism, for functional programmers—shallow parametric polymorphism, for theoreticians it usually means impredicative parametric polymorphism, as used in System F (called also polymorphic or second-order lambda-calculus). The "working programmer" usually identifies this term with inclusion polymorphism, but it does not mean that he did not unconsciously use other kinds of polymorphism (*vide* monsieur Jourdain, who all of his life has spoken prose not being aware of this fact). Our goal is to disclose other kinds of polymorphism which may be found in Java in order to make these mechanisms more accessible to the working programmer.

Since a long time people discussed how to include in Java programming language parametric polymorphism which allows abstracting over types. Some proposals were considered, among them GJ [2]. However it turned out that importing this mechanism from functional programming languages, e.g. Standard ML, raises some problems in connection with inclusion polymorphism. Thorup and Torgersen [13] proposed a variant of bounded polymorphism which integrates the merits of virtual types with F-bounded polymorphism. Their type system has been further developed, formalized, and proven sound by Igarashi and Viroli [10] within the Featherweight GJ calculus [9]. This mechanism in now available in JDK 5.0. Its short description can be found e.g. in [3, 14].

## 2 Varieties of Polymorphism

Many kinds of polymorphism can be found in modern programming languages. Below we provide necessary definitions (somewhat simplified, but suitable for this paper) and shortly characterize polymorphisms discussed in this paper.

*Objects* are programming units that associate data (called instance variables) with the operations (called methods) that can use or affect these data.

*Classes* are extensible templates for creating objects, providing initial values for instance variables and the bodies for methods. New objects can be created from a class with the `new` operator.

In terms of implementation we can recognize *universal polymorphism* when the same code is executed for any admissible type, whereas in case of *ad-hoc polymorphism* different code is executed for each type. There are two major kinds of universal polymorphism: *parametric* and *inclusion polymorphism*, and two major kinds of ad-hoc polymorphism: *coercion* and *overloading*.

*Parametric polymorphism* allows a simple piece of code to be typed "generically", using variables in place of actual types. These type variables are instantiated with concrete types. Parametric polymorphism guarantees uniform behavior on the range of types.

In *inclusion polymorphism* an object can be viewed as belonging to many different classes that need not to be disjoint; that is, there may be inclusion of classes. Inclusion polymorphism models *subtyping* and *subclassing* (*inheritance*).

*Subtyping*. The type of an object is just the set of names and types of its methods. We say type `S` is a *subtype* of `T` (written `S <: T`), if a value of type `S` can be used in any context in which a value of type `T` is expected. We say `T` is a *supertype* of `S` if `S` is a subtype of `T`. Subtyping relation should be reflexive and transitive. Because values may have multiple types in languages supporting subtyping, we say these languages support *subtype polymorphism*. Object types fit naturally into the subtype relation.

*Subclassing* (*inheritance*). Classes are used not only to create objects, but also to create richer *subclasses* by inheritance that allows new classes to be derived from old ones by adding implementations of new methods or *overriding* (i.e. replacing) implementations of old methods. Subclass relation is defined analogically to the subtype relation. We will write `S <: T` to mean also that class `S` is a *subclass* of `T`, which is a *superclass* of `S`. The meaning of "`<:`" should be clear from the context. The reference to an object of a subclass can be used anywhere that a reference to an object of its superclass is expected.

In general we have two hierarchies, one induced by inheritance, the other one corresponding to the subtyping relation. These two hierarchies are, in principle, completely distinct, e.g. in Objective Caml [11], but in many typed object-oriented languages the two hierarchies coincide.

*Bounded polymorphism* integrates parametric and subtype polymorphism, and allows restriction on type variables by specifying upper and/or lower bounds.

*Ad-hoc polymorphism* allows to use the same name for different piece of code that may behave in unrelated ways for each type.

*Coercion* is a semantic operation that converts an argument to the expected type in a situation that would otherwise result in a type error. The special well-known case of coercion is *promotion*.

In *overloading* the same name is used to denote different functions or methods and the context is used to decide which function or method is denoted by a particular instance of the name.

In languages with subtyping we differentiate at least two disciplines of method (or function) selection:
*early binding* is based on static (compile-time) type information;
*late binding* is based on dynamic (run-time) type information.

Classification of type systems can be found in [5], see also [6]. Theoretical foundations of type systems in programming languages are contained in monographs [1, 4, 7, 12].

## 3 Polymorphism "ad hoc"

There are two major kinds of ad-hoc polymorphism: *overloading* and *coercion*, but the boundary between them in many cases is not sharp and depends on the implementation.

### 3.1 Operator Overloading

Operators for arithmetical operations in Java are overloaded for all numeric types and strings, but both arguments must be of the same type. In the expression '`5+3.4`' operator '`+`' denotes addition of two numbers of type `double` and left argument is coerced to the type `double`. In the expression '`5+3.4F`' two numbers of type `float` are added with appropriate coercion of the left argument. Of course different object code is generated by the compiler in both cases. However, one may imagine that operator '`+`' is overloaded for four combinations of argument types (`int` and `double`). The same situation may be observed for other arithmetic operators.

In this example we may consider the same expression to exhibit overloading or coercion (or both), depending on implementation decision.

### 3.2 Method Overloading

Most object-oriented programming languages, including Java, allow programmers to overload method names on classes. A method name is overloaded in a context if it is used to represent two or more distinct methods, and the method represented by the overloaded name is determined by its signature, like the method name `m` from the class `C` defined below. As long as their signatures are different, Java treats methods with overloaded names as though they had completely different names. The compiler statically (using early binding) determines what method code is to be executed.

```
public class C {
   void m( ) { System.out.print("C.m()  "); }
   void m( C other ) { System.out.print("C.m(C)  "); }
}
```

In Java, the overloading can happen when a method in a superclass is inherited in a subclass that has a method with the same name, but different signature.

```
public class SubCOverload extends C {
   void m( SubCOverload other )    // overloading m
   { System.out.print("SubCOverload.m(SubCOverload)  "); }
}
```

Class `SubCOverload` has three overloaded methods with name `m`.

```
public class TestOverload {
  public static void main(String[] args) {
    C c = new C();
    C c1 = new SubCOverload();
    SubCOverload sc = new SubCOverload();
                                          System.out.print(" c: ");
    c.m(); c.m(c); c.m(c1); c.m(sc);      System.out.print("\nc1: ");
    c1.m(); c1.m(c); c1.m(c1); c1.m(sc);  System.out.print("\nsc: ");
    sc.m(); sc.m(c); sc.m(c1); sc.m(sc);  System.out.println();
  }
}
```

Executing code of the class `TestOverload` we obtain the following result.

```
 c: C.m()  C.m(C)  C.m(C)  C.m(C)
c1: C.m()  C.m(C)  C.m(C)  C.m(C)
sc: C.m()  C.m(C)  C.m(C)  SubCOverload.m(SubCOverload)
```

It illustrates the fact, that the class `SubCOverload` has three overloaded methods `m`, and that Java uses early binding for overloaded methods.

### 3.3 Coercion between Primitive Types and Wrapper Classes

The type of an object is just the set of names and of its methods. In Java these types are called *reference types* (since objects in Java are accessed exclusively by references). But Java has also *primitive types* like `int`, `double`, or `boolean`. Values of these types are not objects, but for each primitive type there exists wrapper class, e.g. `Integer`, `Double`, `Boolean`, which can be used in contexts where primitive types are not allowed.

Converting between primitive types, like `int` or `boolean`, and their wrapper classes like `Integer` and `Boolean` was very annoying in old Java. Unfortunately, these back and forth conversions could not be avoided since only objects can be stored in collections. Below the essential part of the wrapper class `Integer` is shown.

```
public final class Integer {
  private int i;
  public Integer(int i) { this.i = i; }
  public int intValue() { return i; }
}
```

The following code illustrates the operations of "wrapping" and "unwrapping" integer value.

```
Integer wi1 = new Integer(5);  // wrapping
int i1 = wi1.intValue();       // unwrapping
```

Using autoboxing/unboxing mechanism the code is much more concise and easier to follow. This is an example of coercion, hence ad-hoc polymorphism.

```
Integer wi2 = 5;  // autoboxing
int i2 = wi2;     // unboxing
```

In old Java (without generic types) the terms *subtype* and *subclass* were basically interchangeable, but in Java 5.0 this relationship is more complicated, as we shall see later.

## 4 Polymorphic Classes

### 4.1 Inclusion Polymorphism

Suppose we want to define a class which provides the basic functionality of a pair without regard for specific types. In old Java this could have been done using inclusion polymorphism. We had to define a pair whose both elements are instances of the `Object` class, since every class in Java inherits from it, hence every object can be stored in such a pair.

```
public class ObjectPair {
  private Object e1;
  private Object e2;

  public ObjectPair(Object e1, Object e2)
  { this.e1 = e1; this.e2 = e2; }
  public Object getFst() { return e1; }
  public Object getSnd() { return e2; }
  public String toString() { return "(" + e1 + ", " + e2 + ")"; }
}
```

Unfortunately, when we use this class, downcasts (with time and memory overhead) are required.

```
ObjectPair p;
p = new ObjectPair("Five", new Integer(5));

String numeral = (String) p.getFst();
Integer number = (Integer) p.getSnd();

p = new ObjectPair(new Integer(5), "Five");
numeral = (String) p.getFst();  // throws ClassCastException
```

Java compiler generates code, which checks dynamically correctness of downcasting operation and possibly throws `ClassCastException`. This affects program efficiency.

**Method overriding with late binding** exhibits quite different behavior than method overloading with early binding. When "object-oriented working programmer" speaks about polymorphism, he usually refers to this mechanism. In Java, when a method is redefined in a subclass with exactly the same signature as the original method in the superclass then we have overriding and the binding of method calls occurs at run time (late binding). If the new method has the same name, but different signature, then we have overloading as described in Subsection 3.2 and the binding occurs at compile time (early binding).

In the example below both classes `SubC1Override` and `SubC2Override` inherit from the class `C`, defined in Subsection 3.2, and they override (redefine) inherited method `m`.

```
public class SubC1Override extends C {
   void m( ) { System.out.print("SubC1Override.m()  "); }
}
```

```
public class SubC2Override extends C {
   void m( ) { System.out.print("SubC2Override.m()  "); }
}
```

In the code below the list of instances of these classes is traversed, and for each object its method `m` is invoked. This method prints the name of a class in which its body has been defined. `LinkedList` is a standard collection, which contains instances of class `Object`. We need downcasting `(C)i.next()` to make use of method `m`. This is a typical program with collections and inclusion polymorphism.

```
import java.util.*;
public class TestOverrideIncl {
   public static void main(String[] args) {
      LinkedList l = new LinkedList();
      l.add(new C());
      l.add(new SubC1Override());
      l.add(new SubC2Override());
      for (Iterator i = l.iterator(); i.hasNext(); ) {
          ( (C)i.next()).m();
      }
      System.out.println();
   }
}
```

Program output is given below. Observe, that the code of the method `m` comes from classes objects are instantiated, and not necessary from class `C`, to which they were downcast. This proves that late binding was used (in C++ terminology these methods are virtual).

```
C.m()  SubC1Override.m()  SubC2Override.m()
```

Unfortunately, downcasting is always dangerous, since there is no guarantee that all objects in the collection are instances of class `C` (or its subclasses).

### 4.2 Parametric Polymorphism

A class for a pair can be defined using parametric polymorphism (or generic class in Java terminology).

```
public class Pair<A, B> {
   private A e1;
   private B e2;

   public Pair(A e1, B e2)
   { this.e1 = e1; this.e2 = e2; }

   public A getFst() { return e1; }
   public B getSnd() { return e2; }
   public String toString() { return "(" + e1 + ", " + e2 + ")"; }
}
```

Below this class is used to create a pair for strings and integers. In Java generic class can be instantiated with reference types only. We do not need downcasting. All type checking is done statically during compilation and `ClassCastException` will never be thrown.

```
Pair<String, Integer> p;
p = new Pair<String, Integer> ("Five", 5);

String numeral = p.getFst();
Integer number = p.getSnd();
```

The code below illustrates the advantages of Java generics over code which uses inclusion polymorphism (class `TestOverrideIncl` from the previous section). Again, there are no casts. Notice also shorter form of the `for` loop.

```
import java.util.*;
public class TestOverrideParam {
   public static void main(String[] args) {
      LinkedList<C> l = new LinkedList<C>();
      l.add(new C());
      l.add(new SubC1Override());
      l.add(new SubC2Override());
      for (C e:l) { // read: for each e of type C in l
          e.m();
      }
      System.out.println();
   }
}
```

This is a typical program with collections and parametric polymorphism.

### 4.3 Integrating Parametric and Inclusion Polymorphism

Java 5.0 integrates parametric and inclusion polymorphism as illustrates class `PairMut`. It extends functionality of class `Pair` with two new methods, which allow to change objects stored in a pair.

```
public class PairMut<A, B> extends Pair<A,B>{
   public PairMut(A e1, B e2) { super(e1, e2); }
   public void setFst(A e) { e1 = e; }
   public void setSnd(B e) { e2 = e; }
}
```

## 5 Implementation of Generics

When generating bytecode for a generic class, Java compiler replaces type parameters by their *erasure*. Basically, erasure gets rid of (or *erasures*) all generic type information, and generates *one* bytecode for a generic class, which contains nothing but ordinary types, and which is executed for each instantiation of this class. This guarantees backward compatibility with legacy code. For an unbounded type parameter its erasure is `Object`. For an upper-bounded type parameter its erasure is the erasure of its upper bound.

With the addition of generics, the relationship between subtyping and subclassing has become more complex. In the code below (references to) objects `pS` and `pI` have distinct types (`Pair<String, String>` and `Pair<Integer, Integer>`, respectively), but are instances of the same class `Pair`, which is called *raw type*. All reference types in old Java were raw types in this terminology and they are legitimate in Java 5.0.

```
Pair<String, String> pS = new Pair<String, String>("fst","snd");
Pair<Integer, Integer> pI = new Pair<Integer, Integer>(1,2);
System.out.println(pS.getClass());        // prints: class Pair
System.out.println(pI.getClass());        // prints: class Pair
System.out.println(pS instanceof Pair);   // prints: true
System.out.println(pI instanceof Pair);   // prints: true
```

Consequently, the expression `pS instanceof Pair<String, String>` is illegal and gives rise to the compilation error *"illegal generic type for instanceof"*.

This erasure implementation enforces some limitations on generics in Java, e.g. type variables in parametric polymorphism cannot be instantiated with primitive types. This limitation is not serious in presence of autoboxing/unboxing mechanism.

For comparison, templates in C++ are generally not type checked until they are instantiated. They are typically compiled into disjoint code for each instantiation rather then a single code, hence problems arise with code bloat, but type variables *can* be instantiated with primitive types.

## 6 Generic Types are not Covariant

We say that type operator `G` is *covariant* (or that subtyping is covariant for `G`) if `S <: T` implies `G<S> <: G<T>`. We call it is *contravariant* if `S <: T` implies `G<T> <: G<S>`. We say that `G` is *invariant* if the conjunction of `S <: T` and `T <: S` implies `G<S> <: G<T>`.

Subtyping is invariant for generic types. We show only why it cannot be covariant. Suppose we want to have a method that prints elements of any pair. Here is a naive attempt using generics:

```
public static void printPairOfObjects(Pair<Object,Object> p) {
   System.out.println( "(" + p.getFst() + ", " + p.getSnd() + ")");
}
```

The problem is that it only works for `Pair<Object,Object>` which is *not* a supertype of all pairs! Compiling the following code we obtain compilation errors in lines 2 and 3.

```
PairMut <String, Integer> pSI;
pSI = new PairMut <String, Integer> ("Five", 5); // 1
printPairOfObjects(pSI);                         // 2
PairMut<Object, Object> pOO = pSI;        // 3: incompatible types
```

Suppose that assignment in line 3 was accepted. Then the following instructions must also be accepted.

```
pOO.setFst(new Object());   // 4
String s = pSI.getFst();    // 5
```

Accessing pair `pSI` through the alias `pOO` arbitrary object can be inserted to the pair, as was done in line 4. Now in line 5 we attempt to assign an `Object` to a `String`! For the same reason line 2 is illegal.

The argument against covariant subtypig for generic classes also applies to arrays, but Java actually permits covariant subtyping of arrays. This feature is now generally considered a flaw in the language design, since it seriously affects the performance of programs involving arrays. The reason is that the unsound subtyping rule must be compensated with a run-time check on *every* assignment to *any* array, to make sure the value being written belongs to the actual type of the elements of the array. If this type checking fails, the `ArrayStoreException` is thrown. In the example below we refer to classes `C` and `SubCOverload` defined in Subsection 3.2. Assignment `arrC = arrSC` is legal, because of covariant subtyping: `SubCOverload <: C` hence `SubCOverload[] <: C[]`.

```
SubCOverload[] arrSC = {new SubCOverload(), new SubCOverload()};
C[] arrC = arrSC;
arrC[0] = new C();
arrSC[0].m(new SubCOverload()); // throws ArrayStoreException
```

Since arrays in Java are covariant, but generic types are invariant, one cannot create an array of a generic type using named variables (but unbounded wildcards are allowed). The declaration:

```
Pair <Integer, Integer>[] iPArr = new Pair <Integer, Integer>[5];
```

is illegal and causes compilation error *"generic array creation"*.

## 7 Wildcards and Bounded Polymorphism

The supertype of all kinds of pairs is `Pair<?,?>`, where wildcard '`?`' stands for some unknown type. Here is the code for our printing method.

```
public static void printPair(Pair<?,?> p) {
   System.out.println("(" + p.getFst() + ", " + p.getSnd() + ")");
}
```

Syntactically, a wildcard is an expression of the form '?', possibly annotated with an upper bound, as in '`? extends T`', or with a lower bound, as in '`? super T`'. As demonstrated in Section 6 subtyping is *invariant* for parameterized types. Using wildcards one may also express *covariant* and *contravariant* subtyping. Parameterized types with `extends`-bounded wildcards give rise to covariant subtyping: if `A<:B`

then `G<? extends A> <: G<? extends B>`. Dually, `super`-bounds give rise to contravariant subtyping: if `A<:B` then `G<? super B> <: G<? super A>`.

Similarly one can specify upper bounds for named type parameters '`S extends Foo`'. In the absence of a type bound, a type parameter is assumed to be bounded by `Object`. Only wildcards can have lower bounds. A named type parameter can have more than one upper bound, but wildcard can have only a single (upper or lower) bound.

## 8 Polymorphic Methods

Methods can also be made generic, independently of the class in which they are defined by adding a list of formal type arguments to its definition. Suppose we want to write static utility method `max`, which returns greater of its two arguments. Static methods are outside the scope of class-level type parameters (this is another limitation caused by erasure semantics), so we have to use generic method, parameterized by a type of its argument. But arguments must be comparable, hence the type must specify appropriate method for comparison, which can be assured by a bound.

Java standard library contains generic interface `Comparable<T>`.

```
public interface Comparable<T> {
  public int compareTo(T o);
}
```

It specifies a method that compares 'this' object with the specified object for order.

Here is our polymorphic method we were looking for:

```
public class Test{
   public static <T extends Comparable<T>> T max(T e1, T e2) {
        return e1.compareTo(e2) > 0 ? e1 : e2;
   }

  public static void main(String[] args) {
        System.out.println(max("Adam", "Dick"));
        System.out.println(max(6, 2));  // autoboxing to Integer
  }
}
```

Notice, that when `extends` is used to denote a type parameter bound, it does not denote a subclass-superclass relationship, but rather a subtype-supertype relationship.

## 9 Conclusions

The notion of polymorphism unifies many concepts present in typed programming languages. We briefly defined some varieties of polymorphism and traced them in Java 5.0. Java shows that "industrial" programming languages are able to express more abstract patterns using rather involved theoretical means, hence the "working programmer" has to be better educated in order to understand them, recognize them in different programming languages under different names and make good use of them. Programmers are often unpleasantly surprised when a mechanism exhibits different behavior in another language, or if different names turn out to denote the same mechanism.

It would be also interesting to compare polymorphic features of some other "industrial" programming languages (e. g. C++, C#) from more abstract perspective than used in manuals or tutorials. "Working programmer" should have a command of many programming languages which quickly evolve, and deeper understanding of underlying concepts would greatly alleviate his task.

## References

1. Abadi M., Cardelli L.: *A Theory of Objects,* Springer-Verlag (1996).
2. Bracha G., Odersky M., Stoutamire D., Wadler P.: *Making the future safe for the past: Adding genericity to the Java programming language,* In: Proceedings of ACM Symposium on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA), ACM Press (1998) pp. 183–200.

3. Bracha, G.: *Generics in the Java programming language,* (2004) `http://java.sun.com/j2se/1.5/pdf/generics-tutorial.pdf`
4. Bruce K. B.: *Foundations of Object-Oriented Languages,* MIT Press (2002).
5. Cardelli L., Wegner P.: *On understandind types, data abstraction and polymorphism,* Computing Surveys **17** (1985) pp. 471–522.
6. Cardelli L.: *Type systems,* In: Tucker A. B., ed.: Handbook of Computer Science and Engineering. CRC Press (1997) pp. 2208–2236.
7. Castagna G.: *Object-Oriented Programming. A Unified Foundation,* Birkhäuser (1997).
8. Cook W. R., Hill W. L., Canning P. S.: *Inheritance is not subtyping,* In: Proceedings of ACM Symposium on Principles of Programming Languages (POPL), ACM Press (1990) pp. 125–135.
9. Igarashi A., Pierce B., Wadler P.: *Featherweight Java: A minimal core calculus for Java and GJ,* In: Proceedings of ACM Symposium on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA), ACM Press (1999) pp. 132–146.
10. Igarashi A., Viroli M.: *On variance-based subtyping for parametric types,* In: Proceedings of European Symposium on Object-Oriented Programming (ECOOP). LNCS **2374,** Springer-Verlag (2002) 441–469.
11. Leroy X.: *The Objective Caml System, release 3.08. Documentation and users manual,* INRIA. (2004) `http://caml.inria.fr`.
12. Pierce B. C.: *Types and Programming Languages,* MIT Press (2002).
13. Thorup K. K., Torgersen M.: *Unifying genericity. Combining the benefits of virtual types and parameterized classes,* In: Proceedings of European Symposium on Object-Oriented Programming (ECOOP). LNCS **1628,** Springer-Verlag (1999) 186–204.
14. Torgersen M., Ernst E., von der Ahé P., Bracha G., Gafter N.: *Adding wildcards to the Java programming language,* In: Proceedings of ACM Symposium on Applied Computing (SAC), ACM Press (2004).

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp. 233-241
ISBN 83-922646-0-6


# Ocena efektywności wdrożenia systemu klasy ERP APS w warunkach ograniczeń wdrożeniowych z zastosowaniem zbiorów rozmytych

Lilianna Ważna*

*Zakład Controllingu i Informatyki Ekonomicznej, Wydział Zarządzania, Uniwesytet Zielonogórski,
l.wazna@wz.uz.zgora.pl

**Streszczenie.** W artykule przedstawiono procedurę oceny efektywności planowanego przedsięwzięcia wdrożeniowego systemu klasy ERP APS (Advanced Planning and Scheduling, Advanced Planning Systems) w średnim przedsiębiorstwie produkcyjnym. Proponowane podejście, uwzględniając aktualny stan przygotowań rozważanego przedsiębiorstwa do wdrożenia wraz z istniejącymi ograniczeniami wdrożeniowymi, pozwala określić, czy wdrożenie danego systemu informatycznego gwarantuje realizację przyjętego przez to przedsiębiorstwo celu. Zastosowanie zbiorów rozmytych pozwala przy tym modelować subiektywne preferencje przedsiębiorstwa wobec systemu oraz związane z przyszłością warunki niepewności.

## 1 Wstęp

Konwencjonalne systemy klasy ERP (Enterprise Resource Planning – Planowanie zasobów przedsiębiorstwa) tylko w ograniczonym zakresie spełniają swoje podstawowe zadanie – efektywnego planowania i sterowania zasobami w przedsiębiorstwie – w szczególności w obszarze produkcji i logistyki. APS (Advanced Planning and Scheduling, Advanced Planning Systems - Zaawansowane planowanie i harmonogramowanie, Systemy zaawansowanego planowania) stanowi nową generację systemów o znacznie udoskonalonej funkcji planowania i sterowania [6]. Oferowane na rynku oprogramowania moduły APS w postaci rozwiązań zintegrowanych w ramach systemu ERP, wykorzystując dane z ERP, umożliwiają optymalizację produkcji i procesów logistycznych, symulacje, oraz planowanie i harmonogramowanie w czasie rzeczywistym, wspierając procesy decyzyjne (por. [4, 6, 8, 12]).

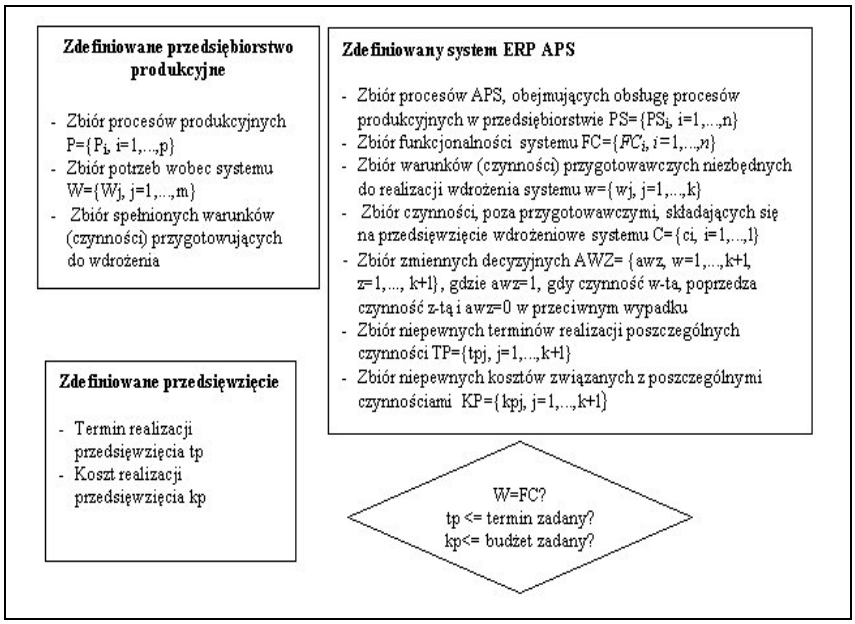
Proces wdrożenia zintegrowanych systemów zarządzania jest inwestycją informatyczną o dużym stopniu złożoności, długotrwałą, kosztowną, wymagającą wielu przygotowań przedsiębiorstwa i dobrej organizacji prac, bez których staje się przedsięwzięciem bardzo ryzykownym (por. [1], [5], [10]). W konsekwencji producenci oferujący systemy ERP APS zmuszeni są do uzależniania zarówno terminu jak i kosztu przedsięwzięcia wdrożeniowego od aktualnego stanu przygotowań danego przedsiębiorstwa. Przedsiębiorstwa produkcyjne natomiast, dążąc do uzyskania określonej poprawy wyników swojej działalności w założonym terminie i budżecie, stają przed koniecznością wdrożenia oraz wyboru odpowiedniego rozwiązania, spośród szerokiej oferty rynkowej. Zachodzi więc potrzeba dokonania takiej oceny efektywności planowanego przedsięwzięcia wdrożeniowego, która uwzględni subiektywne potrzeby i wymagania przedsiębiorstwa wobec systemu oraz aktualny stan przygotowań przedsiębiorstwa do wdrożenia wraz z istniejącymi ograniczeniami wdrożeniowymi.

Proponowane w pracy podejście związane jest więc z opracowaniem oraz implementacją metody, która umożliwi rozwiązywanie następującego problemu decyzyjnego. Dane jest przedsiębiorstwo produkcyjne średniej wielkości o znanych wskaźnikach ilościowych i jakościowych, znanym stanie przygotowań do wdrożenia zintegrowanego systemu informatycznego oraz znanych potrzebach wobec systemu. Dany jest system informatyczny klasy ERP APS o znanych możliwościach funkcjonalnych i wymaganiach technicznych. Poszukiwana jest odpowiedź na pytanie: Czy wdrożenie danego systemu informatycznego gwarantuje określone wartości poprawy wybranych wskaźników efektywności funkcjonowania przedsiębiorstwa w zadanym terminie i budżecie, przy znanych ograniczeniach wdrożeniowych? Celem prezentowanej pracy jest przedstawienie modelu opisanej sytuacji decyzyjnej oraz procedury oceny planowanego przedsięwzięcia wdrożeniowego, stanowiącej podstawę podjęcia decyzji o przyjęciu lub odrzuceniu oferty danego systemu, ze szczególnym uwzględnieniem formalizmu zbiorów rozmytych, służących do opisu zjawisk określonych nieprecyzyjnie i umożliwiających modelowanie subiektywnych preferencji decydenta oraz niepewności związanej z przyszłością.

## 2 Model decyzyjny oceny efektywności wdrożenia systemu ERP APS w przedsiębiorstwie produkcyjnym

W modelu: Przedsiębiorstwo produkcyjne - System klasy ERP APS, zaprezentowanym na rys.1, czynności przygotowawcze oraz pozostałe czynności składające się na planowane przedsięwzięcie oraz ich wzajemne następstwo definiowane są wraz z systemem. W zależności od analizowanych wskaźników efektywności model przyjmuje postać odpowiedniego submodelu.

Wykorzystanie do opisu parametrów niepewnych formalizmu zbiorów rozmytych może stanowić alternatywę dla metody PERT, stosującej przy modelowaniu niepewności rozkłady beta z estymacją za pomocą rozkładów Gaussa. Procedurę postępowania zaprezentowano w rozdziale następnym.




**Rys. 1.** Model Przedsiębiorstwo produkcyjne – System klasy ERP APS

## 3 Procedura oceny efektywności wdrożenia systemu ERP APS w warunkach ograniczeń wdrożeniowych

### 3.1 Zbiory rozmyte w wielokryterialnej ocenie decyzyjnej planowanego przedsięwzięcia wdrożeniowego

Prognozowanie przyszłości nigdy nie gwarantuje wysokiej precyzji podejmowanych decyzji, lecz zwykle wiąże się z pewnymi szacunkami, które już pozwalają je podejmować. Precyzyjne szacunki wyrażone w formie ocen punktowych posiadają bardzo niską wartość, a podejmowanie decyzji jedynie na ich podstawie obarczone jest z reguły dużym błędem. Nie dostarczają bowiem żadnych informacji o związanej z przyszłością niepewności i ryzyku. Służące do opisu zjawisk, określonych nieprecyzyjnie, zbiory rozmyte, stanowiące uzupełnienie lub alternatywę opisu probabilistycznego, uwzględniając brak precyzyjnych informacji, umożliwiają dokonanie opisu niepewności typu subiektywnego, występującej przy podejmowaniu decyzji (por. [7], [14]).

W teorii zbiorów rozmytych zbiór rozmyty $A$ w $X$ charakteryzowany jest funkcją przynależności, która przypisuje elementowi $x$ jego stopień przynależności do zbioru rozmytego $A$ i przyjmuje wartości z przedziału [0,1]. Zbiór rozmyty umożliwia więc płynne określenie stopni, w jakich należą do niego poszczególne elementy. Ten sam element może w różnym stopniu należeć do różnych zbiorów rozmytych. Przykładem zbiorów rozmytych mogą być stosowane w języku naturalnym pojęcia: wysoki, niski, bardzo wysoki, średnio-wy-

soki, duży, mały (por. [11], [14]). Określenia funkcji przynależności można dokonywać na podstawie danych eksperymentalnych lub przy pomocy eksperta.

Szczególny przypadek zbiorów rozmytych stanowią liczby rozmyte, wśród których wyróżnia się liczby rozmyte właściwe i liczby rozmyte płaskie (przedziały rozmyte) (por. [11], [14]). W przypadku wykonywania obliczeń na danych niepewnych mających naturę rozmytą, niepewne wartości rzeczywiste można reprezentować jako liczby rozmyte, nadające się do wyrażania pojęć typu „około 2 tygodni". Stanowi to uzasadnienie wykorzystania ich w niniejszej pracy. Pomimo wielu istniejących podejść do sformułowania operacji na liczbach rozmytych, do celu implementacji informatycznej odpowiednia jest stosowana w pracy metoda reprezentacji liczby rozmytej jako zbioru $\alpha$ -przekrojów i wykonywanie operacji arytmetycznych na $\alpha$ -przekrojach, z wykorzystaniem arytmetyki przedziałów (por. [9], [11], [13], [14]).

Teoria zbiorów rozmytych jest efektywnym środkiem do formułowania zadań oceny wielokryterialnej w warunkach niepewności. Przy dokonywaniu agregacji wielu kryteriów cząstkowych, na równoczesne wykorzystywanie parametrów ilościowych i jakościowych (wyrażonych werbalnie) oraz wprowadzenie ich normalizacji, ułatwiającej dokonanie oceny globalnej pozwala wprowadzenie tzw. kryterium użyteczności, wyrażonego przez funkcję użyteczności (por. [14]). Kryterium użyteczności jest rodzajem zbioru rozmytego, a funkcja użyteczności jest przykładem funkcji przynależności. W celu uwzględnienia stopnia wpływu każdego kryterium cząstkowego na ocenę globalną można określić wagi kryteriów szczegółowych, przez współczynniki względnej ważności, dokonując lingwistycznej oceny ważności porównywania parami i korzystając z macierzy parzystych porównań. Dysponując określonymi wartościami cząstkowych kryteriów użyteczności $\{Ku_i, i=1,...,p\}$ oraz ich wagami $\{QKui, i=1,...,p\}$, do określenia oceny globalnej zastosować można następujące kryteria (por. [2, 3, 13, 14]):

– kryterium addytywne:

$$D = \sum_{i=1}^{p} Q_{Kui} Ku_i , \tag{0}$$

– kryterium multiplikatywne:

$$D = \prod_{i=1}^{p} Ku_i^{Q_{Kuii}} , \tag{2}$$

– kryterium maksymalnego pesymizmu:

$$D = \min\left\{\left(Ku_{1^{Q_{Ku1}}}, \ldots, Ku_{p^{Q_{Kup}}}\right)\right\} \tag{3}$$

### 3.2 Metoda oceny efektywności wdrożenia systemu ERP APS w przedsiębiorstwie produkcyjnym

Przebieg postępowania podzielić można na następujące etapy:

– **Etap 1 – Ocena stanu przygotowań przedsiębiorstwa do wdrożenia systemu**

Etap ten polega na sprawdzaniu szeregu kolejnych warunków, ograniczających realizację planowanego celu i składających się na ocenę aktualnego stanu przygotowań do wdrożenia Do przykładowych warunków należą między innymi: ocena przygotowania infrastruktury technicznej środowiska testowo-rozwojowego, sprzętowo-programowego i teleinformatycznego, kadry przedsiębiorstwa, danych oraz zmian organizacyjnych w przedsiębiorstwie (por. [1], [5]). Każdy nie spełniony warunek jest informacją o przeszkodzie efektywnego wykorzystania oferowanego narzędzia do wyznaczonego celu oraz wskazuje, co jest konieczne do wykonania i w jakim czasie, aby narzędzie mogło przynieść oczekiwany efekt.

– **Etap 2 – Ocena adekwatności funkcjonalności systemu do potrzeb przedsiębiorstwa**

Ten etap dotyczy analizy zestawienia oferty funkcjonalności systemu $\{FCi, i=1,...,n\}$ z wymaganiami i potrzebami przedsiębiorstwa $\{Wj, j=1,...,m\}$. Pozwala określić w jakim stopniu oferta systemu odpowiada wymaganiom przedsiębiorstwa i uwzględnić jego subiektywne preferencje odnośnie wagi tych wymagań. Dokonanie porównania parami poszczególnych wymagań przedsiębiorstwa umożliwia określenie tzw. macierzy parzystych porównań, stanowiącej podstawę wyliczenia współczynników względnej ważności każdego wymagania. Wprowadzenie następnie odwzorowania u: $\{Wj, j=1,..., m\} \rightarrow \{0,1\}$:

$$u(W_j)=\begin{cases}1 & gdy \quad W_j \in \{FC_i, \quad i=1,\ldots,n\} \\ 0 & gdy \quad W_j \notin \{FC_i, \quad i=1,\ldots,n\}\end{cases} \tag{4}$$

oraz wykorzystanie kryterium addytywnego (1) pozwala ustalić stopień adekwatności funkcjonalności systemu do potrzeb przedsiębiorstwa Sadp.

– **Etap 3 – Ocena terminu oraz kosztów realizacji przedsięwzięcia na podstawie danych niepewnych**

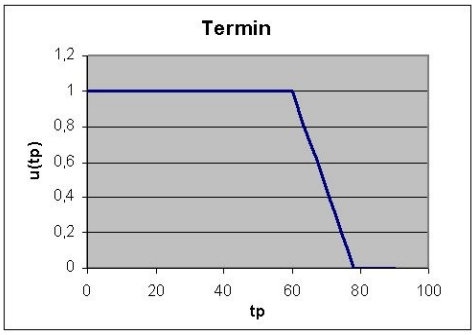
Trzeci etap polega na określeniu na podstawie danych niepewnych, w jakim okresie i przy jakich nakładach finansowych, przy uwzględnieniu ograniczeń z etapu 1, dana inwestycja pozwoli uzyskać określone wartości poprawy wybranych wskaźników efektywności przedsiębiorstwa. Dysponując nieprecyzyjnymi terminami realizacji *{tpj, j=1,...,k+l}* oraz kosztami *{kpj, j=1,...,k}* poszczególnych czynności planowanego przedsięwzięcia i uwzględniając wyniki raportu oceny stanu przygotowań przedsiębiorstwa do wdrożenia, można określić prognozowany termin realizacji przedsięwzięcia *tp* oraz prognozowany koszt przygotowań *kp*. Zastosowano przy tym metodę ścieżki krytycznej, modelowanie parametrów nieprecyzyjnie określonych za pomocą liczb rozmytych oraz operacje sumy i porównania liczb rozmytych z wykorzystaniem $\alpha$ -przekrojów i arytmetyki przedziałów (por. [9], [11], [13], [14]). Uwzględniając ponadto koszt zakupu systemu *kzs* oraz koszt usługi wdrożeniowej *ku* ustalono prognozowaną wartość kosztu przedsięwzięcia *kpc*. Umożliwia to ocenę planowanej inwestycji w dwóch kategoriach: czy zmieści się w planowanym terminie? Czy zmieści się w planowanym budżecie?

– **Etap 4 – Globalna ocena efektywności planowanego przedsięwzięcia**

Etap polegający na agregacji poszczególnych kryteriów cząstkowych, dotyczących oceny użyteczności planowanego przedsięwzięcia. Dla rozważanych kryteriów cząstkowych funkcje użyteczności mogą mieć postać zaprezentowaną na rys.2-4:

– Adekwatność funkcjonalności systemu do potrzeb przedsiębiorstwa - dany system otrzyma pod względem tego kryterium ocenę tym lepszą, im w większym stopniu spełni wymagania przedsiębiorstwa (rys. 2).

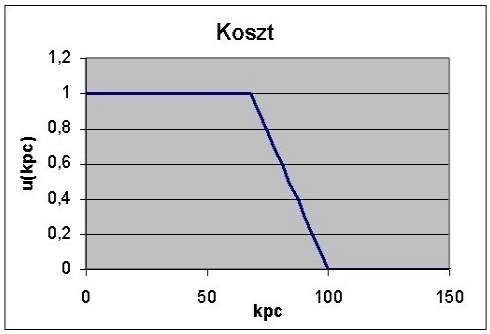

**Rys. 2**. Funkcja użyteczności dla kryterium: Adekwatność



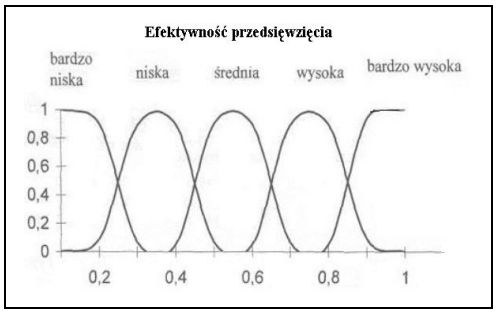

**Rys. 3**. Funkcja użyteczności dla kryterium: Termin

– Termin realizacji przedsięwzięcia – ocena planowanego przedsięwzięcia jest tym gorsza, im bardziej prognozowany termin realizacji przedsięwzięcia przekracza dany termin realizacji planowany przez przedsiębiorstwo (rys. 3: dany termin=60, termin dopuszczalny=78).

– Koszt przedsięwzięcia – im bardziej prognozowany koszt przedsięwzięcia przekracza dany budżet planowany przez przedsiębiorstwo, tym gorsza jest ocena planowanego przedsięwzięcia (rys. 4: dany budżet =68, kwota dopuszczalna=100).




**Rys. 4**. Funkcja użyteczności dla kryterium: Koszt




**Rys. 5.** Wartości oceny efektywności przedsięwzięcia. Definicja własna.

Dysponując określonymi wartościami powyższych cząstkowych kryteriów użyteczności $\{Ku_i, i=1,...,p\}$ oraz ich wagami $\{QKui, i=1,...,p\}$, ustalonymi zgodnie z preferencjami przedsiębiorstwa przy pomocy macierzy parzystych porównań, można określić wartość oceny globalnej planowanego przedsięwzięcia za pomocą kryteriów addytywnego, multiplikatywnego i maksymalnego pesymizmu zgodnie ze wzorami (1)-(3).

Przeprowadzenie powyższej analizy i sprawdzenie przedstawionych warunków daje poszukiwaną ocenę efektywności wdrożenia danego systemu (rys.5). Przyjęto, że przy uzyskaniu efektywności wysokiej i bardzo wysokiej planowane wdrożenie danego systemu informatycznego gwarantuje realizację określonego celu przedsiębiorstwa. Stanowi to rozwiązanie sformułowanego problemu.

### 3.3 Przykład ilustrujący zastosowanie proponowanej procedury

Dane jest przedsiębiorstwo produkcyjne średniej wielkości rozważające w ramach budżetu 68jp. wdrożenie zintegrowanego systemu zarządzania klasy ERP APS w celu zwiększenia wskaźnika wydajności pracy o 7% oraz redukcji zapasów o 10% w terminie 60j. Poszukiwany system powinien umożliwić realizację takich zadań, jak: kontrola kosztów produkcji oraz planowanie potrzeb materiałowych. Ponadto przedsiębiorstwo wymaga specjalnej opieki eksploatacyjnej w ciągu pierwszych 7j. czasu pracy systemu w postaci konsultanta pomocnego w rozwiązywaniu bieżących trudności.

Oferowane jest kompleksowe rozwiązanie klasy ERP dla średnich przedsiębiorstw, jakim jest system proALPHA®APS. Elastyczność systemu pozwala na zakup wybranych elementów spośród następujących modułów: Dane podstawowe, Sprzedaż i dystrybucja, Zakupy, Gospodarka materiałowa, Produkcja, Zarządzanie Projektami, Księgowość finansowa, Księgowość środków trwałych, Rachunek kosztów i wyników. Koszt usługi wdrożeniowej wynosi 18jp i obejmuje usługi konsultanta w początkowym etapie funkcjonowania systemu (10j.).

Planowane przedsięwzięcie składa się z czynności podanych w tabeli 1.

**Tabela 1.** Zbiór czynności składających się na planowane przedsięwzięcie

| Czynność | Nazwa czynności | Czynności poprzedzające |
|---|---|---|
| w1 | Przygotowanie miejsca i zakup serwera | |
| w2 | Modelowanie procesów biznesowych | |
| w3 | Zakup i przygotowanie sieci roboczych, stacji roboczych, drukarek | |
| w4 | Przygotowanie serwera, instalacja systemu | w1 |
| w5 | Konfiguracja i testowanie ystemu | w2, w4 |
| w6 | Szkolenie | w3, w5 |
| c1 | Wdrożenie systemu | w3, w5 |
| c2 | Uruchomienie systemu | c1 |
| c3 | Kontrola i eksploatacja systemu | c3, w5 |

Przebieg zastosowania proponowanej procedury przedstawiono poniżej:

- **Etap 1 – Ocena stanu przygotowań przedsiębiorstwa do wdrożenia systemu**

Wyniki raportu sprawdzania warunków przygotowawczych zawierają informację o spełnieniu przez przedsiębiorstwo warunków: w1 i w3.

- **Etap 2 – Ocena adekwatności funkcjonalności systemu do potrzeb przedsiębiorstwa**

Z zestawienia wymagań przedsiębiorstwa oraz możliwości systemu przedstawionego w tabeli 2, przy uwzględnieniu jednakowych preferencji przedsiębiorstwa wobec poszczególnych wymagań wynika, że proponowana oferta systemu odpowiada w pełni oczekiwaniom przedsiębiorstwa (Sadp=1).

**Tabela 2.** Zestawienie oferty systemu proALPHA®APS z potrzebami przedsiębiorstwa

| Wymagania przedsiębiorstwa | Oferta systemu proALPHA®APS |
|---|---|
| Planowanie potrzeb materiałowych | Gospodarka materiałowa |
| Kontrola kosztów produkcji | Produkcja |
| | Rachunek zysków i strat |
| Opieka eksploatacyjna (7 j.cz.) | Konsultant (10 j.cz.) |

- **Etap 3 – Ocena terminu oraz kosztów realizacji przedsięwzięcia na podstawie danych niepewnych**

Do oceny wykorzystano dane przykładowe dla parametrów nieprecyzyjnych zawarte w tabeli 3.

**Tabela 3.** Przykładowe dane dla parametrów opisujących planowane przedsięwzięcie. Podany wektor zawiera możliwe termniy zakończenia (lub koszty) realizacji wybranej czynności

| Czynność | Termin | Koszt |
|---|---|---|
| w1 | [1,3,2,2,2] | [1,4,1,2,1] |
| w2 | [15,14,16,14,18] | [2,2,4,3,5] |
| w3 | [10,11,14,17,16] | [3,3,4,4,16] |
| w4 | [10,14,11,5,14] | [4,4,3,5,15] |
| w5 | [8,18,14,13,14] | [5,3,6,6,19] |
| w6 | [7,7,6,9,8] | [6,4,7,2,6] |
| c1 | [12,9,10,14,11] | |
| c2 | [8,10,9,11,10] | |
| c3 | [10,8,11,9,9] | |

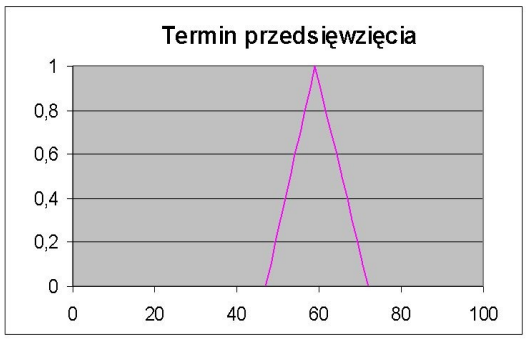
Ponadto koszt zakupu systemu wynosi 30j.

Przy uwzględnieniu wyników raportu z etapu 1, zaproponowanych funkcji użyteczności przedstawionych na rys. 2-4 w analizowanym przykładzie dokonano modelowania danych niepewnych za pomocą trójkątnych funkcji przynależności (por. [9]). Następnie wykorzystano podziały na $\alpha$ -przekroje do wykonania koniecznych operacji arytmetycznych (sumowanie i porównanie) i otrzymano wyniki przedstawione w tabeli 4 i

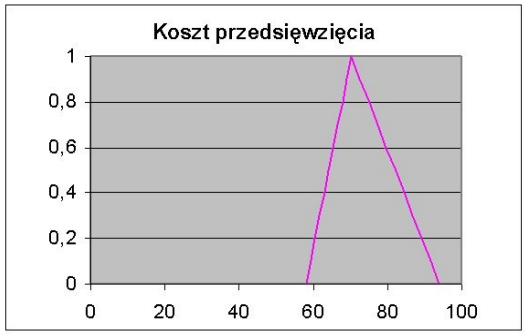
na rys. 6 i 7. Ponieważ w rozważanym przykładzie operacje arytmetyczne sumowania wykonywane były na trójkątnych liczbach rozmytych, wyniki otrzymano również w posatci trójkątnych liczb rozmytych. W ogólnym przypadku w zależności od określonych funkcji przynależności dla modelowania parametrów niepewnych, otrzymane wyniki mogą mieć inną postać zbiorów rozmytych.

**Tabela 4.** Prognozowany termin i koszt planowanego przedsięwzięcia przy modelowaniu niepewności za pomocą zbiorów liczb rozmytych

| tpmin | tpmod | tpmax | kpmin | Kpmod | kpmax | kpcmin | kpcmod | kpcmax |
|---|---|---|---|---|---|---|---|---|
| 47 | 59 | 72 | 10 | 22,2 | 46 | 58 | 70,2 | 94 |



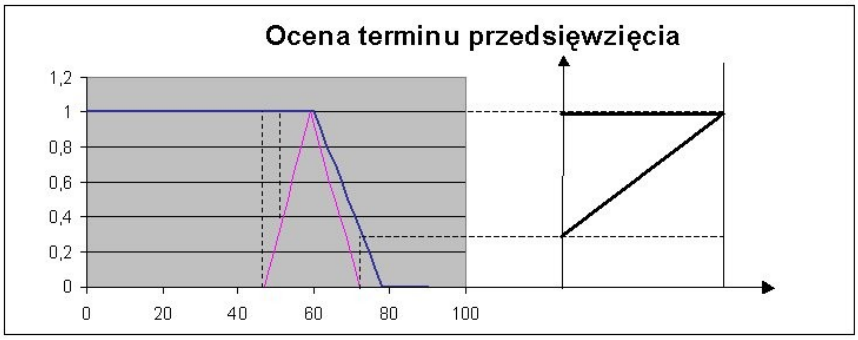

**Rys. 6.** Prognozowany termin przedsięwzięcia



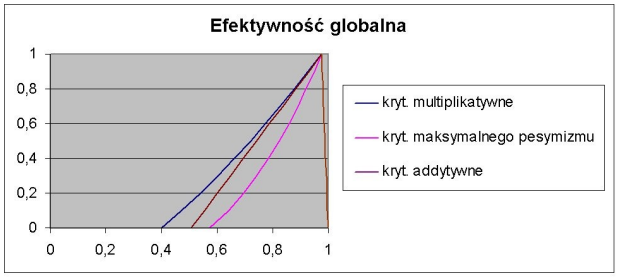

**Rys. 7**. Prognozowany koszt przedsięwzięcia

– **Etap 4 – Globalna ocena efektywności planowanego przedsięwzięcia**

Agregacji przedstawionych kryteriów, jednakowo preferowanych przez rozważane przedsiębiorstwo, dokonano posługując się przedstawionymi wcześniej kryteriami zgodnie ze wzorami (1)-(3). W tym celu otrzymaną wartość prognozowanego terminu przedsięwzięcia oceniono najpierw za pomocą kryterium Termin z rys. 3 (por.[14]). W prawej części rys. 8 znajduje się wynik dokonanej oceny, który ze względu na rozmytą daną wejściową jest również liczbą rozmytą. Analogicznie dokonano oceny według kryterium Koszt i Adekwatność. Następnie wykorzystano podział na $\alpha$ -przekroje i wykonano operacje arytmetyczne zgodnie ze wzorami (1)-(3). Otrzymane wartości globalnej oceny planowanego przedsięwzięcia, dla każdego kryterium (1)-(3), zaprezentowano na rys. 9. Uzyskaną ocenę efektywności można określić jako bardzo wysoką. Zatem z przeprowadzonej analizy wynika, że w rozważanym przypadku wdrożenie danego systemu gwarantuje realizację przyjętego przez przedsiębiorstwo celu, co jest rozwiązaniem postawionego problemu.




**Rys. 8.** Ocena prognozowanego terminu realizacji przedsięwzięcia wg kryterium Termin




**Rys. 9.** Ocena efektywności planowanego przedsięwzięcia wdrożeniowego

## 5. Podsumowanie

Proponowana w pracy procedura daje możliwość oceny planowanego przedsięwzięcia wdrożeniowego pod kątem poprawy wybranych wskaźników efektywności przedsiębiorstwa produkcyjnego, z uwzględnieniem jego obecnego stanu przygotowań do wdrożenia, stwarzając dodatkowe możliwości oceny rentowności projektu informatycznego. Wykorzystanie przy dokonywaniu tej oceny formalizmu zbiorów rozmytych, pozwala na uwzględnianie subiektywnych preferencji decydenta wobec systemu, oraz niepewności, związanej z podejmowaniem decyzji dotyczących przyszłości.

Rozpatrywanie ograniczeń wdrożeniowych przy dokonywaniu tej oceny umożliwia uzyskanie już na etapie planowania odpowiedzi na szereg pytań, takich jak: co jest konieczne do wykonania i w jakim terminie, aby wdrożenie danego narzędzia mogło przynieść oczekiwany efekt? czy przedsiębiorstwo jest przygotowane do realizacji wdrożenia? co stanowi przeszkodę w przeprowadzeniu efektywnego wdrożenia i wykorzystaniu oferowanego narzędzia do realizacji wyznaczonego celu? jaki okres czasu niezbędny jest na realizację nie wykonanych jeszcze prac przygotowawczych? Tymczasem istniejące oceny efektywności ekonomicznej inwestycji nie uwzględniają tego typu ograniczeń.

Ponadto przedstawiona procedura, poprzez identyfikację zależności oczekiwanego efektu od czynników związanych z realizacją przedsięwzięcia, umożliwia kontrolowanie i uaktualnianie prognozy podczas trwania przedsięwzięcia. Implementacja proponowanej metody jest przedmiotem dalszych prac.

## Bibliografia


1. Adamczewski P.: *Zintegrowane systemy informatyczne w praktyce*, Wyd. MIKOM, Warszawa 2000.
2. Dymowa L., Figat P., Zenkowa A.: *Metoda i oprogramowanie do oceny wielokryterialnej i wielopoziomowej decyzji w warunkach niepewności rozmytej*. III Krajowa Konferencja "Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim", Kraków, 19-21 listopada, 2001, s. 575-576. (http://zsiie.icis.pcz.pl/artykuly/pf/metoda.pdf)
3. Figat P.: *Opracowanie metody i oprogramowania do wielokryterialnej i wielopoziomowej oceny alternatyw w zagadnieniach podejmowania decyzji*, Praca magisterska, Częstochowa, 2002. (http://zsiie.icis.pcz.pl/dyplomy/figat.pdf)
4. Gunther H. O., Tempelmeier H.: *Produktion und Logistik* (5. Aufl.) Berlin: Springer, 2003.
5. Grudzewski W. M., Hejduk I. K.:, *Metody projektowania systemów zarządzania*, Wyd. Difin, Warszawa 2004.
6. Kluge P. D., Kuźdowicz P., Andracki S.: *Multi-resource-planning and realtime-optimization based on proALPHA(r) APS solution* W: Automation 2005 : Automatyzacja - nowości i perspektywy : konferencja naukowo-techniczna. Warszawa, Polska, 2005.
7. Kwiesielewicz M.: *Modelowanie niepewności przy użyciu przybliżonych miar prawdopodobieństwa*, Gdańsk 1998.
8. Maciejec L.: *ERP – nowe oblicze*, CIO Magazyn dyrektorów IT, **1**/2005, (http://cio.cxo.pl/artykuly/46129.html)
9. Nikiel G., *Optymalizacja wielokryterialna w projektowaniu procesów wytwarzania – wybrane zagadnienia*, Raport z badań własnych, Bielsko-Biała, 2004, http://www.ath.bielsko.pl/~gnikiel/publik/optym.pdf
10. Patalas J., Banaszak Z.: *Wdrażanie systemów klasy ERP w MSP. Model procesu decyzyjnego*, W: Koncepcje zarządzania systemami wytwórczymi / red. M. Fertsch, S. Trzcieliński,. Poznań: Politechnika Poznańska, 2005, s. 73—79.
11. Piegat A.: *Modelowanie i sterowanie rozmyte*, Akademicka Oficyna EXIT, Warszawa, 1999.
12. Rutkowski K. (red.): *Logistyka on-line. Zarządzanie łańcuchem dostaw w dobie gospodarki elektronicznej*, 2000, PWE.
13. Róg P.: *The method of Intervals Ordering Based on the Probibalistic Approach*, Computer Science, 2002, Vol **I**, N 1, P. 103-108, (2001).
14. Róg P.: *Opracowanie metodyki i oprogramowania do operowania na danych przedziałowych i rozmytych w zastosowaniach modelowania procesów produkcyjnych i podejmowania decyzji*, Praca magisterska, Częstochowa, 2002, http://zsiie.icis.pcz.pl/dyplomy/rog.pdf

XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6
Conference Proceedings pp. 243–254 

# Deterministyczna metoda przetwarzania ciągów danych

Michał Widera*

Instytut Techniki i Aparatury Medycznej ITAM,
ul. Roosevelta 118, 41-800 Zabrze, Polska
michal@widera.int.pl,
strona WWW: http://widera.int.pl

**Streszczenie** W typowych systemach przetwarzania informacji system zarządzania danymi może zostać wydzielony. Niestety, relacyjne systemy zarządzania bazą danych nie są na tyle wydajne, aby podołać zadaniu bieżącego przetwarzania sygnałów w systemie monitorującym. Główny nurt badań nad stworzeniem modelowego systemu zarządzania danymi dla potrzeb systemów monitorujących jest związany ze strumieniowym modelem danych. W większości systemy te funkcjonują jednak niedeterministycznie. W artykule przedstawiono opracowane w ramach prac badawczych deterministyczne metody przetwarzania strumieni danych dla potrzeb zadań przetwarzania sygnałów w medycznych systemach zarządzania danymi. Przedstawiono opracowane w ramach prac badawczych twierdzenia opracowanej algebry ciągów danych (strumieni) wraz z wyprowadzonymi formalnymi dowodami. Udowodniono bezpośredni związek niektórych operatorów z odkrytymi w połowie zeszłego wieku twierdzeniami Beatty oraz Fraenkela.

## 1 Wstęp

Obecnie dostępne systemy zarządzania danymi przetwarzają informacje w oparciu o reguły rachunku relacji. Ten sposób przetwarzania informacji umożliwia opis dowolnych obiektów (Encji) oraz związku pomiędzy nimi (Relacji). Opis struktur danych przy pomocy diagramów Encji-Relacji [1] stał się obowiązującym standardem w dziedzinie projektowania systemów zarządzania danymi.

W typowych systemach przetwarzania informacji system zarządzania danymi może zostać wydzielony. W przypadku systemów, do których dane napływają z zewnętrznych źródeł (np. system monitorujący) relacyjne systemy zarządzania danymi okazały się niedostatecznie wydajne [2]. Poszukiwania aktywnych lub sekwencyjnych [3] rozszerzeń poszerzyły dziedzinę wiedzy nie zapewniając jednak dostatecznie efektywnego rozwiązania problemu. Na przeszkodzie w podniesieniu wydajności stoi utrwalony paradygmat relacyjny [1].

Obecnie główny nurt badań nad stworzeniem modelowego systemu zarządzania danymi dla potrzeb systemów monitorujących jest związany ze strumieniowym modelem danych oraz systemami zarządzania strumieniami danych [4]. Oczekuje się, że zastosowanie go w zadaniu monitorowania pozwoli na prostsze i bardziej eleganckie wyrażenie algorytmów oraz dzięki uproszczeniu konstrukcji uzyskany zostanie znaczący wzrost efektywności przetwarzania informacji.

Podstawową różnicą pomiędzy systemem relacyjnym a strumieniowym jest istnienie w systemie strumieniowym tzw. Ciągłych zapytań [5]. Przez ciągłe zapytanie rozumiemy zapytanie, którego plan realizacji zamknięty jest w martwej pętli. W systemie relacyjnym tego typu funkcjonalność osiąga się poprzez stosunkowo skomplikowany i mało efektywny proces odpytywania bazy danych przez aplikację lub zastosowanie wielu wyzwalaczy. Kolejną różnicą jest sposób wprowadzania danych do systemu. W systemie relacyjnym dane są wprowadzane i zatwierdzane przez użytkownika, natomiast w systemie strumieniowym dane wprowadzane są z zewnętrznych, zazwyczaj autonomicznych podsystemów. Dodatkowo strumień danych ma charakter nieskończony. Tradycyjne systemy zarządzania danymi operują z reguły na danych, dla których oszacowano górną granicę koniecznych zasobów. Operowanie na zbiorach nieskończonych wymaga więc modyfikacji niektórych operatorów.

### 1.1 System zarządzania danymi

Informatyczne systemy monitorujące stosowane w medycynie na bieżąco rejestrują, analizują i prezentują informacje dostarczane przez aparaturę [6]. Celem prowadzonych w prac badawczych było opracowanie systemu zarządzania danymi oraz języka zapytań umożliwiającego formułowanie i realizację zadań

* Praca finansowana ze środków Ministerstwa Nauki i Informatyzacji w latach 2003-2005 jako projekt badawczy 4 T11F 002 25

przetwarzania sygnałów. Opracowano algebrę [7] oraz zaimplementowano język zapytań w systemie zarządzania danymi [8]. Powstałe rozwiązanie umożliwia efektywne przetwarzanie sygnałów dla potrzeb medycznych systemów monitorujących oraz umożliwia prostą realizację i zapis równoległych algorytmów przetwarzania sygnałów.

### 1.2 Projekty systemów strumieniowych

Można obecnie wskazać kilka wiodących projektów badawczych obejmujących zagadnienia systemów strumieniowych [9, 10]. Projekt STREAM [1] jest prowadzony na uniwersytecie Stanford. Projekt Aurora, a obecnie Borealis [11, 12] jest prowadzony na uniwersytecie MIT. Autorzy projektu STREAM poszukują ogólnych reguł przetwarzania strumieni danych w oparciu o rozwinięcie algebry relacyjnej. Dla potrzeb opracowanego modelu zmodyfikowano definicję relacji. Relację $R$ reprezentuje zmienny w czasie zbiór krotek. Zapis relacji przedstawiony został w postaci $R(t)$. Istniejące operatory nazwano relacyjno-relacyjnymi natomiast do operowania na strumieniach danych wprowadzono zespół operatorów relacyjno-strumieniowych i strumieniowo-relacyjnych. Na uwagę zasługuje fakt że nie wprowadzono operatorów strumieniowo-strumieniowych. W przypadku projektu Borealis nie rozpatruje się jawnie zagadnienia połączenia zbioru relacji ze zbiorem strumieni danych, wprowadzone operatory umożliwiają jedynie operacje na strumieniach danych. Projekt prowadzony na uniwersytecie MIT doczekał się komercyjnej implementacji.

## 2 Algebry strumieni danych

Algebra jest jednym z działów matematyki, natomiast pojęcie algebry ogólnej (lub po prostu algebry) dotyczy struktury matematycznej złożonej ze zdefiniowanego zbioru danych oraz pewnej liczby operacji zdefiniowanych na tym zbiorze danych. Przykładem definicji prostej algebry może być następująca struktura matematyczna: $A_1 := (N, (+, -))$. Algebra $A_1$ została zdefiniowana jako zbiór liczb naturalnych na których możemy jednie wykonywać operacje dodawania i odejmowania.

W przypadku algebr opartych na strumieniach danych [11, 4] przyjęto za model danych parę uporządkowaną, której pierwszym elementem jest krotka (krotka - uogólnienie pojęcia pary dla dowolnej liczby elementów) a drugim czas jej wystąpienia - $(s, t)$.

### 2.1 Operacje na strumieniach danych

Prowadząc obliczenia na strumieniach danych zakłada się istnienie danych o następującym charakterze:

- dane nadchodzą w postaci sekwencji elementów,
- każdy element jest przeglądany i przetwarzany tylko raz.

Głównym problemem jest zapewnienie przestrzeni zasobów koniecznych do przetworzenia pojedynczego elementu. Zapotrzebowanie na pamięć operacyjną i czas realizacji zadania w przypadku idealnym powinno być niezależne od wielkości strumienia danych. System wyposażony w ograniczone zasoby uniemożliwia wyznaczenie dokładnej odpowiedzi na zadane zapytanie w oparciu o nieskończony w czasie i wymiarze strumień danych. W takim przypadku za zadowalające przyjmowane są odpowiedzi aproksymowane. Rozważa się kilka podstawowych technik tworzenia algorytmów aproksymujących: szkice, przypadkowe próbkowanie, histogramy oraz algorytmy falkowe.

W przypadku zadań przetwarzania sygnałów konieczne jest przedstawienie algorytmów deterministycznych, z tego powodu przedstawione założenia systemów strumieniowych nie mogą zostać bezpośredniego zastosowane w realizacji tych zadań.

## 3 Przetwarzanie ciągów danych

Potrzebę deterministycznego przetwarzania strumieni danych dostrzeżono w trakcie prac nad systemami czasu rzeczywistego. Twórcy rozwijający projekt QStream [13] określili założenia umożliwiające stworzenie systemu zarządzania danymi dla potrzeb tego rodzaju systemów. Zaprezentowane wyniki prac badawczych [14, 7] idą o krok dalej. Zaproponowano deterministyczną algebrę strumieni danych opartą na matematycznych podstawach teorii liczb a w szczególności układów pokrywających (ang. *Covering System*). Rozważanym problemem jest sposób wyznaczania podziału zbioru dodatnich liczb naturalnych. Dwie sekwencje dokonują podziału zbioru liczb naturalnych jeśli ich przekrój jest zbiorem pustym, natomiast ich suma tworzy zbiór liczb naturalnych. Formalne sformułowanie twierdzenia będącego podstawą badań nad tym problemem podał S. Beatty w 1926r [15].

**Twierdzenie 1 (Beatty).** *Jeśli $p, q$ są dodatnimi liczbami niewymiernymi i zachodzi pomiędzy nimi zależność $\frac{1}{p} + \frac{1}{q} = 1$ to sekwencje $\{\lfloor np \rfloor\}_{n=1}^{\infty} = \lfloor p \rfloor, \lfloor 2p \rfloor, \lfloor 3p \rfloor, \ldots$ oraz $\{\lfloor nq \rfloor\}_{n=1}^{\infty} = \lfloor q \rfloor, \lfloor 2q \rfloor, \lfloor 3q \rfloor, \ldots$ dokonują podziału zbioru dodatnich liczb całkowitych.*

Podstawą prowadzonych rozważań w literaturze jest następująca sekwencja nazywana sekwencją Beatty (podłoga).

$$\mathcal{B}(\alpha, \alpha') := \left( \left\lfloor \frac{n - \alpha'}{\alpha} \right\rfloor \right)_{n=1}^{\infty} \tag{1}$$

lub sekwencją Beatty (sufit):

$$\mathcal{B}^{(c)}(\alpha, \alpha') := \left( \left\lceil \frac{n - \alpha'}{\alpha} \right\rceil \right)_{n=1}^{\infty} \tag{2}$$

Gdzie $\alpha$ oznacza gęstość, $\frac{1}{\alpha}$ oznacza nachylenie, $\alpha'$ oznacza przesunięcie oraz $\frac{-\alpha'}{\alpha}$ y-przechywcenie.

Twierdzenie Beaty umożliwia podział zbioru jedynie w oparciu dobór dwóch liczb niewymiernych. W 1969 Aviezri S. Fraenkel [16] uogólnił poprzednie twierdzenie przedstawiając sposób tworzenia sekwencji dokonujących podziału zbioru również w oparciu o liczby wymierne.

**Twierdzenie 2 (Fraenkel).** *Sekwencje $\mathcal{B}(\alpha, \alpha')$ oraz $\mathcal{B}(\beta, \beta')$ dokonują podziału zbioru $\mathbb{N}$ wtedy i tylko wtedy gdy następujące pięć warunków zostanie spełnionych:*

1. $0 < \alpha < 1$.
2. $\alpha + \beta = 1$.
3. $0 \leqslant \alpha + \alpha' \leqslant 1$.
4. *Jeśli $\alpha$ jest liczbą niewymierną, wtedy $\alpha' + \beta' = 0$ i $k\alpha + \alpha' \notin \mathbb{Z}$ dla $2 \leqslant k \in \mathbb{N}$.*
5. *Jeśli $\alpha$ jest liczbą wymierną, (niech $q \in \mathbb{N}$ będzie najmniejszą liczbą taką że $q\alpha \in \mathbb{N}$ ) wtedy $\frac{1}{q} \leqslant \alpha + \alpha'$ i $\lceil q\alpha' \rceil + \lceil q\beta' \rceil = 1$.*

Dowód twierdzenia można odnaleźć w literaturze [17]. Twierdzenia przedstawione w dalszej części pracy opierają się na tych badaniach. Przedstawione metody znajdują obecnie zastosowanie w zadaniu przetwarzania sygnałów [18].

### 3.1 Model danych

Przyjęty w literaturze [19, 4, 20] model danych $(s, \tau)$ , umożliwia opis zjawisk, które w tej samej chwili czasu przyjmują kilka różnych stanów. Jest to rozwiązanie nadmiarowe w stosunku do potrzeb opisu strumienia danych, w którym nie występują dwie różne krotki w tej samej chwili czasu. Takie założenie znacząco komplikuje definicje wprowadzanych operacji [21]. Strumień danych dla potrzeb strumieniowego systemu zarządzania danymi opisany jest [7, 22] przez parę uporządkowaną $(s_n, \Delta_n)$, której pierwszym elementem jest ciąg krotek a drugim ciąg wartości wyznaczających ciąg różnic czasu pomiędzy kolejnymi krotkami.

Model danych w postaci $(s_n, \tau_n)$ opisuje tą samą klasę zjawisk co model $(s, \tau)$. Natomiast model różnicowy $(s_n, \Delta_n)$ uniemożliwia przedstawienie strumienia danych, w którym występują dwie różne krotki w tej samej chwili czasu $\tau$.

Zawężając przedział zastosowania modelu przyjęto [7], że ciąg odstępów czasu pomiędzy kolejnymi krotkami jest wartością stałą. Dzięki temu można określić model danych w następujący sposób: $(s_n, \Delta)$. Taki model danych odpowiadający serii czasowej jest podstawą dalszych rozważań.

Przyjmuje się że relacje oznaczane są literami alfabetu łacińskiego [23]. Dla strumieni przyjęto podobną konwencję. Dodatkowo z każdym strumieniem danych związana jest wartość oznaczająca stały odstęp czasu pomiędzy kolejnymi krotkami. Wartość tą zapisujemy po nazwie strumienia danych po przecinku, np. zapis A,3 oznacza strumień danych A którego krotki przychodzą raz na trzy sekundy. Z każdym strumieniem związane jest pojęcie schematu.

Schemat strumienia to dowolny, uporządkowany zbiór atrybutów opisujących kolejne pola krotek należących do strumienia danych. Analogicznie jak w przypadku modelu relacyjnego, każdy atrybut odpowiada kolumnie danych. Przykładowy identyfikator strumienia w postaci A(imię ,nazwisko),3 oznacza strumień A, którego każda krotka składa się z dwóch atrybutów: imię i nazwisko oraz każda kolejna krotka pojawia się raz na trzy sekundy. Przez schemat jednorodny strumienia danych rozumiemy schemat, którego wszystkie atrybuty są tego samego typu.

### 3.2 Wprowadzone operatory

Po analizie większości typowych operacji prowadzonych na danych otrzymywanych z urządzeń medycznych zidentyfikowano następujący zbiór operacji: Przeplot, Rozplątanie, Suma, Różnica, Projekcja, Selekcja, Agregacja i Serializacja (AGSE) oraz Przesunięcie. Przyjmując, że dane są dwa strumienie A, B operacje zostały zdefiniowane w następujący sposób:

**Suma i różnica** W przypadku łączenia strumieni danych pochodzących z urządzeń pracujących synchronicznie występuje potrzeba złączenia strumieni danych niejako w naturalny sposób, poprzez złączenie ich schematów. O ile w przypadku strumieni danych, których elementy nadchodzą z identyczną częstością problem sumarycznego złączenia nie sprawia trudności, o tyle w przypadku strumieni danych, których elementy nadchodzą z różną częstością, elementy strumienia nadchodzącego z mniejszą częstością muszą zostać powielone.

Uwzględniając różnice pomiędzy zbiorem relacji oraz zbiorem strumieni danych operację złączenia sumarycznego (sumy) i rozłączenia różnicowego (różnicy) zdefiniowano w następujący sposób: wartością wyrażenia $\Sigma(A,B)$ jest strumień danych. Sposób tworzenia kolejnych krotek strumienia wynikowego oraz wyznaczania wartości $\Delta$ tego strumienia opisuje wzór:

$$c_n = \begin{cases} a_n | b_{\left\lfloor \frac{n\Delta_a}{\Delta_b} \right\rfloor} & \Delta_c = \Delta_a \\ a_{\left\lfloor \frac{n\Delta_b}{\Delta_a} \right\rfloor} | b_n & \Delta_c = \Delta_b \end{cases}, \Delta_c = \min(\Delta_a, \Delta_b) \tag{3}$$

Gdzie:
$\Delta_{a,b,c}$ to wartości określające stały odstęp czasu pomiędzy krotkami w strumieniach $A$,$B$ oraz $C$.
$n$ oznacza pozycję kolejnej krotki
$a_n, b_n, c_n$ oznczają krotki strumieni $A$, $B$ oraz $C$.
Część całkowita liczby rzeczywistej $x$ oznaczana jest jako $\lfloor x \rfloor$. Sufit $\lceil x \rceil$ liczby rzeczywistej $x$ to najmniejsza liczba całkowita nie mniejsza od $x$.

*Przykład 1.* Zakładając istnienie dwóch strumieni danych:
`Alfa(znak),2: (1,2,3,4,...)` oraz `Beta(znak),1: (a,b,c,d,e,f,g,...)`
to wyrażenie: $\Sigma(Alfa, Beta)$ opisuje strumień danych o postaci:
`Omega(znak,znak),1: ((1,a),(1,b),(2,c),(2,d),...)`

Operacja różnicy umożliwia odtworzenie strumieni biorących udział w operacji sumowania. Należy jednak podkreślić, że nie jest to operacja projekcji [1, 23]. Krotki, które w wyniku sumowania strumieni danych o różnych częstościach zostały powielone, są w wyniku operacji rozłączenia różnicowego usuwane. Dopiero zastosowanie operatora projekcji umożliwia odtworzenie pierwotnej postaci strumienia danych.

Argumentem tej operacji jest strumień danych, (który w domyśle został poddany operacji sumowania) oraz para dwóch liczb, przedstawiających stosunek odstępów czasu dwóch pierwotnych strumieni danych. Sposób tworzenia kolejnych krotek strumienia wynikowego opisuje:

$$a_n = \begin{cases} c_n & \Delta_b \geqslant \Delta_a \\ c_{\left\lceil \frac{n\Delta_a}{\Delta_b} \right\rceil} & \Delta_b < \Delta_a \end{cases} \tag{4}$$

Symboliczny zapis tej operacji przedstawia się następująco $\delta_{a,b}(S)$. Gdzie $a$,$b$ oznaczają parę liczb określających stosunek odstępów czasu pierwotnych strumieni danych. $S$ jest strumieniem, argumentem operacji różnicy.

*Przykład 2.* Zakładając istnienie strumienia danych Omega o postaci przedstawionej poprzednio to wyrażenie $\delta_{1,2}(Omega)$ opisuje strumień danych o postaci:
`Alfa2(znak,znak),2: ((1,a),(2,c),(3,e),...)`

**Przeplot i rozplątanie** Operacje sumarycznego splątania (przeplotu) i rozplątania różnicowego (rozplątania) tworzą ortogonalny zbiór operatorów w stosunku do operacji sumy i różnicy. O ile w przypadku sumy i różnicy schematy łączonych strumieni danych mogą być różne, o tyle w przypadku operacji przeplotu i rozplątania schematy łączonych strumieni danych muszą być ze sobą zgodne. Dwa schematy strumieni danych są ze sobą zgodne jeśli odpowiadające sobie kolejne typy pól są takie same.

Jeśli częstość obu strumieni jest identyczna, wynikowy strumień danych po przeplocie zawiera naprzemian krotki ze strumieni będących argumentami operacji. Jeśli częstość argumentów jest różna to kolejność, w jakiej umieszczane są po sobie krotki wyznacza reguła (5) określająca operację przeplotu.

Operacja ta zapisywana jest w następujący sposób $\phi(A, B)$. Następujący wzór przedstawia sposób wyznaczania kolejnych krotek strumienia wynikowego:

$$c_n = \begin{cases} b_{n-\lfloor nz \rfloor} & \lfloor nz \rfloor = \lfloor (n+1)\, z \rfloor \\ a_{\lfloor nz \rfloor} & \lfloor nz \rfloor \neq \lfloor (n+1)\, z \rfloor \end{cases}, z = \frac{\Delta_b}{\Delta_a + \Delta_b}, \Delta_c = \frac{\Delta_a \Delta_b}{\Delta_a + \Delta_b} \tag{5}$$

**Twierdzenie 3.** *Operacja splątania zapewnia sekwencyjne pokrycie obu zbiorów zawierających elementy strumieni danych.*

*Dowód. W pierwszym kroku analizowany jest pierwszy warunek (równości) w równaniu (5). Dla każdego n spełniającego warunek*$^{(*)}$ $\lfloor nz \rfloor = \lfloor (n+1)z \rfloor$*, kolejne wartości wyrażenia* $n - \lfloor nz \rfloor$ *tworzą drugi (skojarzony) ciąg liczb naturalnych wybierających kolejne elementy z ciągu* $b_n$*.*
*Czyli dla każdego n spełniającego warunek*$^{(*)}$ *powinna zachodzić dla wyrażenia* $n - \lfloor nz \rfloor$ *zależność* $x_n = x_{n+1} - 1$*. Formalnie zapis tego zdania przedstawia się następująco:*

$$n - \lfloor nz \rfloor = (n+1) - \lfloor (n+1)\, z \rfloor - 1 \tag{6}$$

*Podstawiając warunek*$^{(*)}$ *do powyższej zależności otrzymujemy:*

$$n - \lfloor (n+1)z \rfloor = (n+1) - \lfloor (n+1)\, z \rfloor - 1 \tag{7}$$

*Poprzez proste algebraiczne uproszczenie dochodzimy do następującej tożsamości:* $n = (n+1) - 1$
*Drugą część dowodu (w oparciu o warunek nierówności) prowadzi się analogicznie.* □

*Przykład 3.* Przy założeniu istnienia strumieni danych `Alfa` oraz `Beta`, wyrażenie $\phi(Alfa, Beta)$ opisuje przeplot dwóch strumieni danych tworzących strumień wynikowy o postaci:
`Gamma(znak),2/3: (a,b,1,c,d,2,e,f,3,...)`
Natomiast wyrażenie $\phi(Beta, Alfa)$ opisuje strumień wynikowy o postaci:
`Delta(znak),2/3: (1,a,b,2,c,d,3,e,f,...)`
Jak widać operacja przeplotu nie jest przemienna .

Rozplątanie jest operacją komplementarną w stosunku do operacji przeplotu. Rozplątując strumień można wybrać te krotki ze strumienia, które należały do jednego z argumentów operacji splątania. Argumentami operacji są: strumień danych oraz liczba określająca odstęp czasu. Dla formalności przyjęto dwa operatory określające tę operację: $\Theta_n(A)$ oraz $\sim \Theta_n(A)$. Pierwszy z nich przedstawia operację uzyskania pierwotnego strumienia danych, a drugi „reszty" z operacji rozplątania. Operację rozplątania definiują dwa wzory:

$$a_n = c_{n+\left\lceil \frac{(n+1)\Delta_a}{\Delta_b} \right\rceil}\ ,\ \Delta_a = \frac{\Delta_c \Delta_b}{|\Delta_c - \Delta_b|} \tag{8}$$

oraz

$$b_n = c_{n+\left\lfloor \frac{n\Delta_b}{\Delta_a} \right\rfloor},\ \Delta_b = \frac{\Delta_c \Delta_a}{|\Delta_c - \Delta_a|} \tag{9}$$

**Twierdzenie 4.** *Operacja rozplątania spełnia postulaty twierdzenia Fraenkela.*

*Dowód.* W pierwszej części dowodu poszukiwana jest sekwencja Beatty przedstawiona w sposób (9) zaprezentowany w definicji operacji rozplątania. Zapis przy pomocy tej notacji przedstawia się następująco:

$$\left( n + \left\lfloor \frac{nb}{a} \right\rfloor \right)_{n=1}^{\infty} \tag{10}$$

Jeśli $n \in \mathbb{N}$ to wtedy na mocy własności (21) można powyższe równanie zapisać:

$$\left( \left\lfloor \frac{n - \alpha'}{\alpha} \right\rfloor \right)_{n=1}^{\infty} = \left( \left\lfloor n + \frac{nb}{a} \right\rfloor \right)_{n=1}^{\infty} \tag{11}$$

Upraszczając,

$$\left( \left\lfloor n\alpha^{-1} - \frac{\alpha'}{\alpha} \right\rfloor \right)_{n=1}^{\infty} = \left( \left\lfloor n\frac{a+b}{a} \right\rfloor \right)_{n=1}^{\infty} \tag{12}$$

Symbol $-\frac{\alpha'}{\alpha}$ oznacza y-przechwycenie, jeśli wartość przesunięcia sekwencji $\alpha' = 0$ wtedy $\alpha = \frac{a}{a+b}$ , czyli sekwencja (10) może zostać przedstawiona w następujący sposób:

$$\mathcal{B}\left(\frac{a}{a+b}, 0\right) := \left(\left\lfloor n\frac{a+b}{a}\right\rfloor\right)_{n=1}^{\infty} \tag{13}$$

Poprzez kilka prostych przekształceń algebraicznych z sekwencji opisującej wybór kolejnych krotek w operacji rozplątania (9) otrzymano postać sekwencji Beatty. W kolejnej części dowodu na mocy postulatów twierdzenia Fraenkela określamy residuum sekwencji (10).

1. Wartość wyrażenia $\alpha = \frac{a}{a+b}$ dla $a, b > 0$ jest mniejsza od jedności i większa od zera.
2. Warunek $\alpha + \beta = 1$ jest spełniony dla $\beta = \frac{b}{a+b}$.
3. Dla $\alpha' = 0$ postulat jest równoważny 1.
4. Poszukujemy rozwiązań w zbiorze liczb wymiernych.
5. Jeśli $q\alpha \in \mathbb{N}$ i $q \in \mathbb{N}$ oraz $\frac{1}{q} \leqslant \alpha + \alpha'$ to skoro $\alpha' = 0$ to $\frac{1}{q} \geqslant \frac{a}{a+b}$ a to jest prawdą dla $q \leqslant \frac{a+b}{nwd(a,b)}$ z tego wynika że $\left\lceil \frac{a+b}{nwd(a,b)}\beta' \right\rceil = 1$. Czyli $\beta' = \frac{nwd(a,b)}{a+b}$.

W wyniku przekształceń otrzymamy postać ciągu przedstawioną w równaniu (9) opisującym operację rozplątania. Postać ciągu residuum dla $\mathcal{B}\left(\frac{a}{a+b}, 0\right)$ spełniająca postulaty twierdzenia Fraenkela przedstawia się następująco:

$$\mathcal{B}\left(\frac{b}{a+b}, \frac{nwd(a,b)}{a+b}\right) = \left(\left\lfloor \frac{(n+1) - \frac{nwd(a,b)}{a+b}}{\frac{b}{a+b}} \right\rfloor\right)_{n=1}^{\infty} \tag{14}$$

Opuszczając nawiasy opisujące sekwencję po kilku prostych przekształceniach można wykazać, że:

$$\left\lfloor \frac{(n+1) - \frac{nwd(a,b)}{a+b}}{\frac{b}{a+b}} \right\rfloor := \left\lfloor n\frac{a}{b} + n + \frac{a}{b} + 1 - \frac{nwd(a,b)}{b} \right\rfloor \tag{15}$$

Przyrównując wyznaczoną postać residuum do przyjętej sekwencji (8) otrzymujemy równanie którego prawdziwość należy udowodnić:

$$\left\lfloor n\frac{a}{b} + n + \frac{a}{b} + 1 - \frac{nwd(a,b)}{b} \right\rfloor = n + \left\lceil \frac{(n+1)a}{b} \right\rceil \tag{16}$$

Stąd można wykazać, że:

$$\left\lfloor \frac{(n+1)a}{b} - \frac{nwd(a,b)}{b} \right\rfloor + 1 = \left\lceil \frac{(n+1)a}{b} \right\rceil \tag{17}$$

Podstawiając za $n+1$ liczbę naturalną $n$ otrzymujemy:

$$\left\lfloor n\frac{a}{b} - \frac{nwd(a,b)}{b} \right\rfloor + 1 = \left\lceil n\frac{a}{b} \right\rceil \tag{18}$$

W dalszej części dowodu zastosowanie znajdą następujące własności operacji podłoga oraz sufit:

$$\lfloor x \rfloor = \lceil x \rceil \Leftrightarrow x \in \mathbb{N} \tag{19}$$

$$\lfloor x \rfloor + 1 = \lceil x \rceil \Leftrightarrow x \in \mathbb{R} \backslash \mathbb{N} \tag{20}$$

$$\lfloor x + C \rfloor = \lfloor x \rfloor + C \Leftrightarrow c \in \mathbb{N} \tag{21}$$

Uwzględniając własności $nwd(a,b)$ dowód można rozważać w oparciu o następujące dwa przypadki:

$$nwd(a,b) = b \Leftrightarrow \frac{a}{b} = c \in \mathbb{N} \tag{22}$$

oraz

$$1 \leqslant nwd(a,b) \leqslant a \Leftrightarrow 0 < \frac{a}{b} < 1 \tag{23}$$

Uwzględniając przypadek (22), równanie (18) można zapisać w postaci:

$$\left\lfloor \frac{(n+1)a}{b} - \frac{b}{b} \right\rfloor + 1 = \left\lceil \frac{(n+1)a}{b} \right\rceil \tag{24}$$

Uwzględniając własności (19,21) oraz dziedzinę dla tego przypadku (22) można stwierdzić, że obie sekwencje tworzą te same elementy.

Rozważając kolejny przypadek (23) można założyć że istnieją takie dwie liczby $a$ i $b$, dla których równanie (18) nie jest fałszywe, czyli dla:

$$n\frac{a}{b} - \frac{nwd(a,b)}{b}, n\frac{a}{b} \in \mathbb{N} \tag{25}$$

Korzystając z własności (19,21) nie zachodzi:

$$\left\lfloor n\frac{a}{b} - \frac{nwd(a,b)}{b} + 1 \right\rfloor \neq \left\lceil n\frac{a}{b} \right\rceil \tag{26}$$

Korzystając z własności (19) poszukiwane są takie $a$ i $b$ dla których zachodzi własność:

$$n\frac{a}{b} - \frac{nwd(a,b)}{b} + 1 \neq n\frac{a}{b} \tag{27}$$

Równanie (18) jest spełnione dla $nwd(a,b) = b$, a w rozważanej dziedzinie (23) $1 \leqslant nwd(a,b) \leqslant a$ równanie to nie ma rozwiązań. Nie istnieją takie $a$ i $b$ należące do dziedziny (25) które by przeczyły rozważanemu równaniu (18).

Analizując drugą własność (20) można założyć że istnieją dwie takie liczby $a$ i $b$ że równanie (18) nie jest spełnione. Czyli dla $a$ i $b$ takich, że:

$$n\frac{a}{b} - \frac{nwd(a,b)}{b}, n\frac{a}{b} \in \mathbb{R}\backslash\mathbb{N} \tag{28}$$

Powinna zawsze zachodzić:

$$n\frac{a}{b} - \frac{nwd(a,b)}{b} \neq n\frac{a}{b} \tag{29}$$

Nie istnieją jednak dwie takie liczby, dla których $nwd(a,b) = 0$

Dla $\frac{a}{b} \in \mathbb{R}\backslash\mathbb{N}$ równanie to jest zawsze prawdziwe.

Tak więc oba równania (8,9) są przypadkiem sekwencjami Beatty spełniającymi postulaty twierdzenia Fraenkela dla liczb wymiernych. □

*Przykład 4.* Zakładając istnienie przedstawionego w poprzednim przykładzie strumienia Gamma,2/3 to wyrażenie $\Theta_1(Gamma)$ opisuje strumień danych:

```
Alfa(znak),2: (1,2,3,4,...)
```

Natomiast wyrażenie $\Theta_2(Gamma)$ opisuje strumień danych

```
Beta(znak),1: (a,b,c,d,e,f,g,...)
```

**Rzutowanie** Operator rzutowania (projekcji) wykorzystywany jest przy tworzeniu nowego strumienia, który powstaje ze strumienia $A$ poprzez usunięcie lub zmianę kolejności kolumn. Wartością wyrażenia $\pi_{T_1,T_2,\ldots,T_n}(A)$ jest strumień, który powstaje ze strumienia $A$ poprzez przepisanie wartości wszystkich atrybutów $T_1, T_2, \ldots, T_n$. Operacja rzutowania nie wpływa na wartość stałego odstępu czasu pomiędzy kolejnymi krotkami w strumieniu. Wartość $\Delta$ po operacji rzutowania pozostaje niezmieniona.

*Przykład 5.* Zakładając istnienie przedstawionego strumienia danych Alfa2 wyrażenie $\pi_{Znak}(Alfa2)$ opisuje strumień danych o postaci

```
Alfa(znak),2: (1,2,3,...)
```

**Selekcja** W wyniku zastosowania operatora selekcji na strumieniu $A$ powstaje nowy strumień, do którego należy pewien podzbiór krotek strumienia $A$. Krotki strumienia wynikowego są wybierane według kryterium określonego przez pewien warunek $W$. Tę operację, bardzo podobnie jak w przypadku algebry relacji, oznaczono symbolem $\sigma_{W,\delta}(A)$. Schemat strumienia wynikowego po operacji selekcji pozostaje bez zmian. Porządek atrybutów również zostaje zachowany. $W$ stanowi wyrażenie warunkowe oznaczające prawdę lub fałsz. Operandami mogą być stałe lub atrybuty strumienia $A$. Jeśli wyrażenie przyjmuje wartość prawda, to wówczas krotka jest dołączana do strumienia wynikowego, w przeciwnym wypadku

krotka nie zostanie dołączona. Operacja selekcji wpływa bezpośrednio na wartość stałego odstępu czasu pomiędzy kolejnymi krotkami w strumieniu. Wartość $\Delta$ strumienia wynikowego jest określana z góry i przyjmuje wartość $\delta$.

*Przykład 6.* Zakładając istnienie strumienia Omega(znak,znak),1 wyrażenie:
$\sigma_{is\_number(znak),2}(Omega)$ opisuje strumień Alfa(znak),2.

**Agregacja i serializacja** Akronim AGSE powstał ze złożenia pierwszych sylab słów Agregacja oraz Serializacja. Jest to operacja, której argumentem jest jeden strumień danych o jednorodnym schemacie. W wyniku tej operacji powstaje kolejny strumień danych o zadanym schemacie oraz innej, wyznaczonej wartości $\Delta$. Operację oznacza się symbolem $\Psi_{n,\lambda}(A)$. Symbol $\lambda$ oznacza krok, z jakim przesuwane jest ruchome okno danych nad strumieniem. Jest on wyrażony w liczbie krotek. Symbol $n$ oznacza szerokość tego okna danych wyrażoną w ilości atrybutów schematu danych.

*Przykład 7.* Jeśli istnieje strumień Alfa(znak),1 wyrażenie $\Psi_{2,2}(Alfa)$ opisuje nowy strumień o postaci: $Agse1(znak, znak), 2$ którego elementy tworzą ciąg: $((a, b), (c, d), (e, f), \dots)$ . Natomiast wyrażenie $\Psi_{1,1}(Beta)$ opisuje strumień Beta(znak),1. Pierwsza operacja jest przykładem agregacji, natomiast druga jest przykładem serializacji.

**Przesunięcie** Operator przesunięcia to podstawowy operator stosowany w zapisie filtrów sygnałowych. Umożliwia on odwołanie się do danych archiwalnych. Jest to operacja w wyniku, której tworzony jest strumień danych, którego elementy są przesunięte względem argumentu o zadaną liczbę krotek. Operację oznaczono symbolem $\tau_n(A)$. Symbol $n$ oznacza liczbę krotek, o jaką należy opóźnić strumień wynikowy w stosunku do argumentu $A$.

*Przykład 8.* Jeśli istnieje strumień Alfa(znak),1 wyrażenie: $\tau_2(Alfa)$ definiuje strumień
`Shift2(znak),1: (3,4,5,6,...)`

## 4 Własności operacji

W przypadku optymalizacji czasowej planów realizacji zapytań stosuje się m.in. regułę przenoszenia operacji selekcji ponad operacje złączeń [24]. Podobne zagadnienia są rozważane w przypadku systemów strumieniowych [25]. W trakcie prac nad deterministycznym sposobem optymalizacji czasowej odkryto kilka własności oraz metod algebraicznego upraszczania wyrażeń strumieniowych. Pierwsza z własności dotyczy zaburzenia kolejności elementów. Następna określa stosunkowo prostą zasadę przemienności sumowania. Ostatnia metoda, określona jako metoda dopasowania przeplotu umożliwia zamianę kolejności atrybutów operacji przeplotu przy pewnych warunkach.

### 4.1 Zaburzenie kolejności zdarzeń

**Twierdzenie 5.** *Kolejność elementów w strumieniu nie zawsze odzwierciedla faktyczną kolejności występowania elementów w świecie rzeczywistym*

*Dowód.* Tą własność można w prosty sposób odkryć, analizując operację przeplotu zastosowaną na następujących strumieniach danych:
`Alfa(znak),2: (1,2,3,4,5,6...)`
`Epsilon(znak),3: (a,b,c,d,e,f....)`
Wyrażenie $\phi(Epsilon, Alfa)$ opisuje przeplot strumieni danych o postaci:
`Tau(znak),6/5: (1,2,a,3,b,4,5,c,6,d,...)`
W strumieniu Tau krotka oznaczona literą c występuje po krotce oznaczonej cyfrą 5. Tymczasem w strumieniu Epsilon krotka oznaczona literą c występuje w 9 sekundzie, a w strumieniu Alfa krotka oznaczona cyfrą 5 występuje w 10 sekundzie. Naturalny porządek zdarzeń został w strumieniu wynikowym naruszony. Należy wyciągnąć następujący wniosek: Kolejność krotek w strumieniu danych nie zawsze odpowiada kolejności ich występowania w świecie rzeczywistym. Prowadząc analizę względem czasu zawartego w strumieniach danych konieczne jest uwzględnienie tego aspektu i zastosowanie w tym przypadku operacji rozplątania w celu uzyskania pierwotnej postaci strumieni danych. □

### 4.2 Przemienność sumowania

**Twierdzenie 6.** *Operacja sumowania strumieni danych, nie uwzględniając kolejności atrybutów sumowanych strumieni danych jest przemienna.*

*Dowód.* Załóżmy, że: $C = \Sigma(A, B)$ oraz $C = \Sigma(S, A)$. Korzystając ze wzoru na sumę strumieni danych można przedstawić dwie zależności:

$$C = \Sigma(A,B) \quad c_n = \begin{cases} a_n | b_{\lfloor \frac{n\Delta_a}{\Delta_b} \rfloor} & \Delta_a < \Delta_b \\ a_{\lfloor \frac{n\Delta_b}{\Delta_a} \rfloor} | b_n & \Delta_a > \Delta_b \end{cases} \tag{30}$$

oraz

$$D = \Sigma(S,A) \quad d_n = \begin{cases} s_n | a_{\lfloor \frac{n\Delta_s}{\Delta_a} \rfloor} & \Delta_s < \Delta_a \\ s_{\lfloor \frac{n\Delta_a}{\Delta_s} \rfloor} | a_n & \Delta_s > \Delta_a \end{cases} \tag{31}$$

Pomijając kolejność atrybutów wynikającą z operacji połączenia elementów krotek oraz zmieniając kolejność warunków można wyprowadzić następujący wzór:

$$D = \Sigma(S,A) \quad d_n = \begin{cases} a_n | s_{\lfloor \frac{n\Delta_a}{\Delta_s} \rfloor} & \Delta_s > \Delta_a \equiv (\Delta_a < \Delta_s) \\ a_{\lfloor \frac{n\Delta_s}{\Delta_a} \rfloor} | s_n & \Delta_s < \Delta_a \equiv (\Delta_a > \Delta_s) \end{cases} \tag{32}$$

Podstawiając we wzorze (32) za symbol $S$ symbol $B$ dowodzimy poprawności twierdzenia o przemienności operacji sumowania strumieni danych. Istnieje jeszcze jeden, nieuwzględniony przypadek - kiedy $\Delta$ obu strumieni danych jest równa. Dowód nie został zaprezentowany z uwagi na trywialność. □

### 4.3 Metoda dopasowania przeplotu

Jak pokazano na przykładzie operacji przeplotu, nie jest to operacja przemienna. Jednak istnieje algebraiczna metoda umożliwiająca zmianę kolejności atrybutów operacji przeplotu przy pewnych założeniach.

**Twierdzenie 7.** *Jeśli wybierzemy dwie liczby naturalne i,k, których stosunek jest równy stosunkowi wartości elementów $\Delta_n$ strumieni łączonych operacją przeplotu to operacja przeplotu strumieni danych przesuniętych względem tych wartości tworzy strumień danych równy strumieniowi powstałemu poprzez przeplot, przesunięcie względem sumy wartości tych liczb i zamianę kolejności argumentów tej operacji. Formalnie: Jeśli spełnione są następujące warunki: $\frac{i}{k} = \frac{\Delta_a}{\Delta_b}$ oraz $i, k \in \mathbb{N}$ to:*

$$\phi(\tau_i(A), \tau_k(B)) = \tau_{i+k}(\phi(B, A)) \tag{33}$$

*Dowód.* Analizując lewą stronę równania (33) oraz biorąc pod uwagę równanie opisujące operację przeplotu (5) można otrzymać następujący wzór:

$$C = \phi(\tau_i(A), \tau_k(B)) \quad c_n = \begin{cases} b_{(n-\lfloor nz \rfloor)+i} & \lfloor nz \rfloor = \lfloor (n+1)z \rfloor \\ a_{(\lfloor nz \rfloor)+k} & \lfloor nz \rfloor \neq \lfloor (n+1)z \rfloor \end{cases} \tag{34}$$

Analizując prawą stronę równania (33) oraz biorąc pod uwagę równanie opisujące operację przeplotu (5), można otrzymać następujący wzór:

$$C = \tau_{i+k}(\phi(B, A)) \quad c_n = \begin{cases} a_{\lfloor (n+i+k)z \rfloor} & \lfloor nz \rfloor = \lfloor (n+1)z \rfloor \\ b_{n+i+k-\lfloor (n+i+k)z \rfloor} & \lfloor nz \rfloor \neq \lfloor (n+1)z \rfloor \end{cases} \tag{35}$$

Rozważając warunki, dla których równania dobierają odpowiednie próbki ze strumieni $A$ i $B$ oraz korzystając z założeń twierdzenia (33) możemy stwierdzić, że:

$$b_{n-\lfloor nz \rfloor+i} = b_{n+i+k-\lfloor (n+i+k)z \rfloor} \Longrightarrow -\lfloor nz \rfloor = k - \lfloor (n+i+k)z \rfloor \tag{36}$$

oraz:

$$i + k = \frac{\Delta_a}{\Delta_b}k + k = k\left(\frac{\Delta_a}{\Delta_b} + 1\right) = \frac{k}{z} \tag{37}$$

Mając na uwadze założenia tezy (33) oraz poprzednie równania można przedstawić następujący wniosek:

$$-\lfloor nz \rfloor = k - \lfloor k + nz \rfloor \tag{38}$$

Biorąc pod uwagę założenie tezy (33) tzn. $k \in \mathbb{N}$ można stwierdzić że równanie (38) jest spełnione. Druga część dowodu, prowadzona w oparciu o warunek nierówności, jest analogiczna. □

## 5 Budowa planów zapytań

W poprzednich punktach przedstawiono zbiór operatorów oraz metody umożliwiające poprawę efektywności dostęp do danych. Efektywnością realizacji zarządza procesor zapytań. Procesor zapytań jest elementem systemu zarządzania danymi, który służy do przetłumaczenia zapytań wyrażonych w języku formalnym na ciąg operacji na bazie danych. Błędna strategia realizacji zapytań może prowadzić do powstania algorytmów, które przetwarzają zapytania w znacznie dłuższym czasie niż jest to niezbędne. W większości relacyjnych systemów zarządzania danymi w celu wyrażenia wewnętrznej reprezentacji zapytań zapisanych w języku SQL stosowana jest notacja algebry relacji. Zaletą algebry relacji jest możliwość reprezentacji wyrażeń algebraicznych w postaci drzewa wyrażeń opisujących logiczny plan zapytania [24].

Podobne zasady przyjęto podczas tworzenia procesora zapytań dla potrzeb tworzonego systemu. W celu wyrażenia wewnętrznej reprezentacji zapytań stosowana jest przedstawiona w pracy algebra strumieni danych. Zdefiniowany dla potrzeb systemu zbiór operacji różni się zbioru operacji zdefiniowanych w algebrze relacji jednak podobnie jak w przypadku algebry relacji przewidziano możliwość reprezentacji wyrażeń algebraicznych w postaci drzewa wyrażeń. Aby stworzyć drzewo wyrażeń procesor zapytań wykonuje szereg etapów [24]. Najpierw należy przeanalizować składniowo zapytanie zapisane w deklaratywnym języku zapytań. Do opisu języków programowania stosuje się gramatyki formalne, a w szczególności gramatyki bezkontekstowe [26]. Pozwalają one na przejrzystą reprezentacje struktury zapytania w postaci drzewa rozbioru składniowego oraz umożliwiają stworzenie algorytmu, który będzie akceptował poprawne zdania. Następnie należy przekształcić drzewo składniowe w drzewo wyrażeń algebraicznych wg przyjętej algebry danych. Na tym etapie system zarządzania danymi decyduje o sposobie wykonania zapytania wybierając najbardziej efektywny sposób realizacji zadania. Stworzone drzewo wyrażeń nazywane jest też logicznym planem zapytania. W ostatniej fazie logiczny plan zapytania należy przekształcić w fizyczny plan zapytania, który wskazuje nie tylko wykonywane operacje, ale również kolejność ich wykonywania oraz algorytm stosowany do realizacji każdej z nich, a także sposoby przekazywania danych pomiędzy kolejnymi krokami algorytmu.

### 5.1 Drzewa wyrażeń

Jedno wyrażenie algebraiczne może zawierać wiele operatorów. Działają one kaskadowo na wynikach poprzednich działań. Zatem można przedstawić kolejność wykonywania operatorów w postaci drzewa wyrażeń. Liśćmi są nazwy strumieni danych, każdy węzeł wewnętrzny jest etykietowany operatorem.

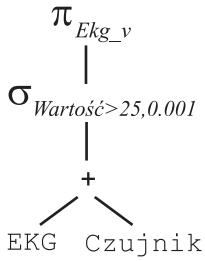
Załóżmy ze są dane dwa strumienie danych: `EKG( pacjent_id, ekg_v ), 0.001` oraz `Czujnik( pacjent_id, wartość ),0.1`. Pierwszy strumień danych zawiera informacje o zmierzonych potencjałach elektrycznych serca, natomiast drugi strumień zawiera pomiary z czujnika biometrycznego. Zapytanie, którego zadaniem jest stworzenie strumienia danych zawierającego jedynie próbki EKG dla których pole wartość strumienia czujnik jest większe od 25.

```
SELECT ekg_v
STREAM bol_ekg
FROM EKG + Czujnik
FILTER BY wartość > 25, 0.001
```

Tak sformułowane zapytanie jest tłumaczone przez analizator składniowy na logiczny plan zapytania, w którym pierwszy krok polega na połączeniu strumieni `EKG` oraz `Czujnik`. W następnym kroku wykonywana jest selekcja, odpowiadająca klauzuli `FILTER BY`, a ostatni krok to rzutowanie na listę atrybutów umieszczoną w klauzuli `SELECT`. Wyrażenie algebraiczne powyższego zapytania przedstawiono w postaci drzewa na rysunku 1.

Istnieje kilka wyrażeń równoważnych wyrażeniu przedstawionemu na rysunku 1. Równoważnych w tym sensie, że bez względu na wybór planu zapytania ich wyniki są takie same. Ogólna zasada znana z algebry relacyjnej stanowi, że (zazwyczaj) warto wykonywać selekcje jak najszybciej. Podobna zasada obowiązuje w przypadku przetwarzania strumieni danych gdyż inne operacje trwają zazwyczaj dłużej niż selekcje, które redukują rozmiary strumieni danych.

Zastosowanie przez procesor zapytań wprowadzonej metody dopasowania przeplotu (33) oraz przemienności sumowania (30,31) umożliwia procesorowi zapytań dobór bardziej odpowiedniego planu realizacji zapytania.




**Rysunek 1.** Logiczny plan zapytania

## 6 Podsumowanie

W ramach prac badawczych realizowane są zadania mające na celu stworzenie systemów monitorujących dla medycyny [6]. W większości przypadków zadanie zarządzania danymi powierza się bezpośrednio systemowi monitorowania. Takie rozwiązane początkowo nie komplikuje znacząco konstrukcji systemu. Jednak z czasem sytuacja ulega zmianie. Okazuje się, że zaimplementowane algorytmy są silnie związane z wybraną implementacją i dlatego zastosowanie nowszych rozwiązań sprzętowych jest utrudnione. Prowadzone prace mają na celu przedstawienie systemu zarządzania danymi wyposażonego w formalny język zapytań, który rozwiąże ten problem. Rozwiązanie oparto na rozszerzonej koncepcji strumieniowego przetwarzania danych. Zastosowanie proponowanego systemu upraszcza konstrukcję algorytmów przetwarzania sygnałów, zapewnia bezpieczeństwo danym oraz rozwiązuje problemy z współużytkowaniem informacji w systemie monitorującym.

Systemy zarządzania strumieniami danych to obecnie jedną z bardziej intensywnie rozwijanych dziedzin nauki związanej z problematyką systemów baz danych. Prowadzone prace mają na celu wykazanie efektywności zastosowania stworzonego modelu przetwarzania informacji w stosunku do istniejących rozwiązań [27]. Zadanie to jest o tyle utrudnione, że przyjęty model relacyjny jest modelem uniwersalnym. Od modelu strumieniowego oczekuje się, że zastosowanie go w zadaniu monitorowania pozwoli na prostsze i bardziej eleganckie wyrażenie zapytań oraz dzięki uproszczeniu konstrukcji uzyskany zostanie znaczący wzrost efektywności przetwarzania informacji. Osiągnięcie stopnia uniwersalności modelu relacyjnego będzie jednak bardzo utrudnione.

W artykule przedstawiono opracowane w ramach prac badawczych twierdzenia algebry ciągów danych (strumieni) wraz z wyprowadzonymi dowodami. Udowodniono bezpośredni związek zastosowanych operatorów z odkrytymi w połowie zeszłego wieku twierdzeniami Teorii Liczb. Twierdzenia te doczekały się stosunkowo niedawno kolejnych dowodów [17].

## Literatura

1. C.J. Date. *Wprowadzenie do systemów baz danych.* Wydawnictwa Naukowo-Techniczne, 2000.
2. Michał Widera, Adam Domański, and Piotr Kasprzyk. Analiza zastosowania baz danych w zadaniu przetwarzania sygnałów biomedycznych. In *Bazy danych Modele, technologie, narzędzia - Analiza danych i wybrane zastosowania*, pages 371–378. WKiŁ, 2005.
3. Alberto Lerner and Dennis Shasha. Aquery: Query language for ordered data, optimization techniques, and experiments. In *Proceedings of 29th International Conference on Very Large Data Bases*, pages 345–356, 2003.
4. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16. ACM Press, 2002.
5. Shivnath Babu and Jennifer Widom. Continuous queries over data streams. *SIGMOD Rec.*, 30(3):109–120, 2001.
6. Michał Widera, Adam Matonia, Janusz Wróbel, Janusz Jeżewski, Krzysztof Horoba, and Tomasz Kupka. Neonatal surveillance system based on data stream technology. In *Journal of Medical Informatics & Technologies JMIT*, volume 9, pages 93–106. Dept. of Computer Systems University of Silesia, October 2005.
7. Michał Widera, Janusz Jeżewski, Ryszard Winiarczyk, Janusz Wróbel, Krzysztof Horoba, and Adam Gacek. Data stream processing in fetal monitoring system: I. algebra and query language. In *Journal of Medical Informatics and Technologies*, volume 5, pages 83–90. Dept. of Computer Systems University of Silesia, 2003.
8. Michał Widera, Janusz Wróbel, Adam Widera, and Adam Gacek. System zarządzania danymi dla potrzeb medycznych systemów monitorujących. In *Współczesne problemy Systemów Czasu Rzeczywistego*, pages 448–458. WNT, 2004.

9. Michał Widera and Stanisław Kozielski. Strumieniowe systemy zarządzania danymi przegląd rozwiązań. In *Bazy Danych Modele, metody formalne, bezpieczeństwo*, pages 257–266. WKiŁ, 2005.
10. Michał Widera and Stanisław Kozielski. Metody przetwarzania w strumieniowych systemach zarządzania danymi. In *Bazy Danych Modele, metody formalne, bezpieczeństwo*, pages 267–276. WKiŁ, 2005.
11. Hari Balakrishnan, Magdalena Balazinska, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Eddie Galvez, Jon Salz, Michael Stonebraker, Nesime Tatbul, Richard Tibbetts, and Stan Zdonik. Retrospective on aurora. *The VLDB Journal*, 13(4):370–383, December 2004.
12. Daniel J Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong-Hyon Hwang, Wolfgang Lindner, Anurag S Maskey, Alexander Rasin, Esther Ryvkina, Nesime Tatbul, Ying Xing, and Stan Zdonik. The design of the borealis stream processing engine. In *Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, Asilomar, CA, January 2005.
13. Sven Schmidt, Henrike Berthold, and Wolfgang Lehner. Qstream: Deterministic querying of data streams. In *Proceedings of the 2004 VLDB Conf.*, pages 1365–1368, 2004.
14. Michał Widera, Janusz Wróbel, Aleksander Owczarek, Adam Matonia, and Michal Jezewski. Data management system for computer aided biophysical monitoring. In *Proc. 27th Annual Int. Conf.on the IEE-EMBS, 2005 Innovation from Biomolecules to Biosystems*, pages 1.6.2–5. IEEE EMBC '05, IEEE Press, September 2005.
15. S. Beatty. Problem 3173. *Amer. Math. Monthly*, 33:159, 1926.
16. A. S. Fraenkel. The bracket function and complementary sets of integers. *Canad. J.Math*, 21:6–27, 1969.
17. Kevin O'Bryant. Fraenkel's partition and brown's decomposition. *The Electronic Journal of Combinatorial Number Theory INTEGERS*, 3:A11–17, 2003.
18. S. Samadi, M.O. Ahmad, and M.N.S. Swamy. Characterization of nonuniform perfect-reconstruction filter-banks using unit-step signal. *IEEE Transactions on Signal Processing*, 52(9):2490–2499, 2004.
19. Lukasz Golab and M. TamerOzsu. Issues in data stream management. *SIGMOD Rec.*, 32(2):5–14, 2003.
20. Daniel J. Abadi, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora: a new model and architecture for data stream management. *The VLDB Journal*, 12(2):120–139, 2003.
21. Arvind Arasu and Jennifer Widom. A denotational semantics for continuous queries over streams and relations. *SIGMOD Rec.*, 33(3):6–11, 2004.
22. Janusz Wróbel, Michał Widera, Krzysztof Horoba, Janusz Jeżewski, and Ryszard Winiarczyk. Declarative algebra and continuous query language for biomedical stream processing in fetal monitoring system. In *proc. 26 th International Conference of IEEE Engineering in Medicine and Biology Society*, pages 3175–3178. IEEE IFMBE, 2004.
23. Jeffrey D. Ullman and Jennifer Widom. *Podstawowy wykład z systemów baz danych.* Wydawnictwa Naukowo-Techniczne, 2001.
24. Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Implementacja systemów baz danych.* Wydawnictwa Naukowo-Techniczne, 2003.
25. M. Tamer and Ozsu. Processing sliding window multi-joins in continuous queries over data streams. In *VLDB*, pages 500–511, 2003.
26. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Kompilatory. Reguły, metody i narzędzia.* Wydawnictwa Naukowo-Techniczne, 2002.
27. Arvind Arasu, Mitch Cherniack, Eduardo F. Galvez, David Maier, Anurag Maskey, Esther Ryvkina, Michael Stonebraker, and Richard Tibbetts. Linear road: A stream data management benchmark. In *Proceedings of the 2004 VLDB Conf.*, pages 480–491, 2004.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp.255-265
ISBN 83-922646-0-6


# Context-aware security and secure context-awareness in ubiquitous computing environments

Konrad Wrona and Laurent Gomez

SAP Research,
805, Avenue du Docteur Maurice Donat, 06250 Mougins, France
konrad.wrona@sap.com, laurent.gomez@sap.com

Context-awareness is emerging as an important element of future wireless systems. In particular, concepts like ambient intelligence and ubiquitous computing rely on context information in order to personalize services provided to their end users. However, security implications of employing context-awareness in computing systems are not well understood. Security challenges in context-aware systems include integrity, confidentiality and availability of context information, as well as end user's privacy. Another interesting and open question is to what extent availability of additional context information could be used in order to optimise and reconfigure security-related services themselves.

## 1. Introduction

Ubiquitous computing is referring to scenarios in which computing is omnipresent, and particularly in which devices that are traditionally perceived as dumb are endowed with computing capability [1]. In ubiquitous computing, context information (such as user's location, time, etc.) can have a strong impact on application adaptation. We consider that application adaptation is performed at both application logic and security management levels. At the application logic level, an application can, e.g., modify contrast of the screen depending on the surrounding brightness. At the security management level, an application can, e.g., adapt an encryption mechanism used for communication with remote application depending on the nature of network over which data will be exchanged. We start with presenting the state of the art in context-aware computing in Section II.

Processing of context information introduces two major challenges. The first one is related to a heterogeneous character of context information: context can come from different context information provider and can be of different kinds. A thermometer delivers temperature whereas user's device provides its IP address in a wireless network. The challenge raised by the diversity of context is discussed in Section III, where we present an initial taxonomy of context information. The second challenge is security in context-aware systems. In Section IV, we describe new challenges and opportunities for information security arising in context-aware environments, and in particular in ubiquitous computing environment. In Section V we present possible ways of using context information to influence security services, in particular in order to manage security and trust in context-aware systems. Finally, in Section VI we present one of the ongoing European research projects focusing on security in context-aware systems.

## 2. Context-aware computing

As more advanced mobile applications begin to emerge, we can observe an increasing interest in use of context information in mobile environment [2], [3], [4]. Several terminologies and classifications for context information have been proposed. Section II-A provides a comprehensive set of definitions related to context-aware computing, which we use in our research. In Section II-B, we describe a life-cycle of a context in a mobile application environment.

### 2.1. Terminology

As context information we understand any kind of information, which can be used to characterize the state of an entity. An entity might be any kind of asset of a computing system such as user, software, hardware, media storage or data [5]. Moreover, we define as context information provider (CIP) any kind of entity which delivers context information. A context information provider can be for example a thermometer, a GPS receiver or a watch. In this paper, we call a system context-aware if it uses any kind of context information before or during service provisioning, including, e.g., service design, implementation, and delivery. The smart floor system is a good illustration of a context-aware system [6]. The smart floor is a device equipped with force measuring sensors, so that it can detects users walking on it. The purpose of such system can be to identify users. In such case, the context-aware system is the backend system to which users authenticate themselves; the context information is the force pressure measured by the smart floor; and the smart floor acts as the context information provider.

### 2.2 Life-cycle of Context

We assume that a context information provider delivers context information to a context-aware system following the life-cycle illustrated in Figure 1. The main steps in a life-cycle of context information are:

- **Discovery of context information**: In this step, a context-aware system discovers available context information providers. The discovery can be performed either in a push or a pull mode [7], i.e. the context-aware system can actively look for CIPs or can passively receive information about available CIPs.
- **Acquisition of context information**: In this step, a context-aware system collects context information from the discovered context information providers and stores it in a context information repository for further reasoning. Similar to the process of discovery, the acquisition is performed either in a pull or a push mode. In a pull mode, the context-aware system explicitly requests for context information whereas in a push mode, context information providers push context information to the context-aware system. For example, a GPS receiver can either push every second a geographical position to a context-aware system or the context-aware system can pull the GPS receiver every second for the geographical position.
- **Reasoning about context information**: reasoning mechanisms enable applications to take advantage of the available context information. The reasoning can be performed based on a single piece of context information or on a collection of such information. For example, in the case of a context-aware e-Health application, user's health status can be evaluated based on both his heart rate and blood pressure provided by medical sensors.


**Fig. 1.** Context life cycle.

### 2.3. Context Architecture

Several architectures have been developed for supporting discovery, acquisition and reasoning about context information. Two particularly interesting approaches are represented by the Context Toolkit Architecture [8] and the Context Broker Architecture (CoBrA) [9].

The Context Toolkit Architecture aims at facilitating development and deployment of context-aware applications. The basic components of this architecture are context widgets, interpreters, aggregators and a discoverer. Context widgets are software components that provide access to context information. Aggregators collect all the relevant context information for a specific application. Interpreters provide abstraction of context information by reasoning about them. Finally, a discoverer aims at determining what context can be currently sensed from the environment. Application can either get access to aggregator, interpreter or a widget. Moreover,

the Context Toolkit enables basic access control for privacy protection. Figure 2 gives an overview of the Context Toolkit Architecture.

The Context Broker Architecture (CoBrA) proposes architecture for discovery, acquisition and reasoning about context information. It includes also mechanisms for privacy protection of context information. CoBrA assumes that all context information providers have a prior knowledge about the broker's presence and that they communicate with the context broker via a standardized protocol. CoBrA is based on a Web Semantic Ontology, which is described in Section III. CoBrA architecture is depicted in Figure 3.

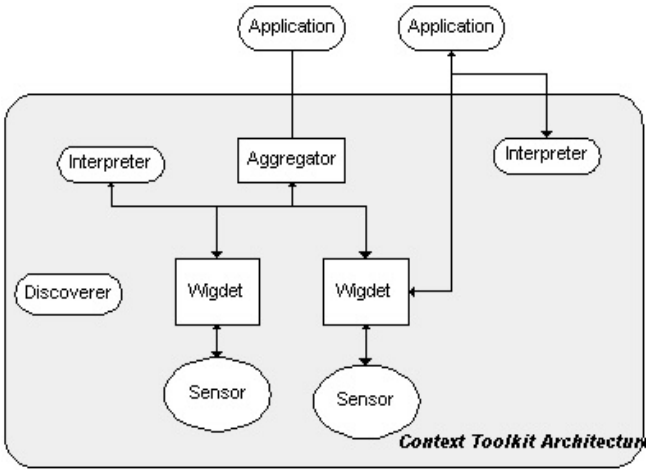



**Fig. 2.** Context Toolkit Architecture (arrows show the flow of context information).

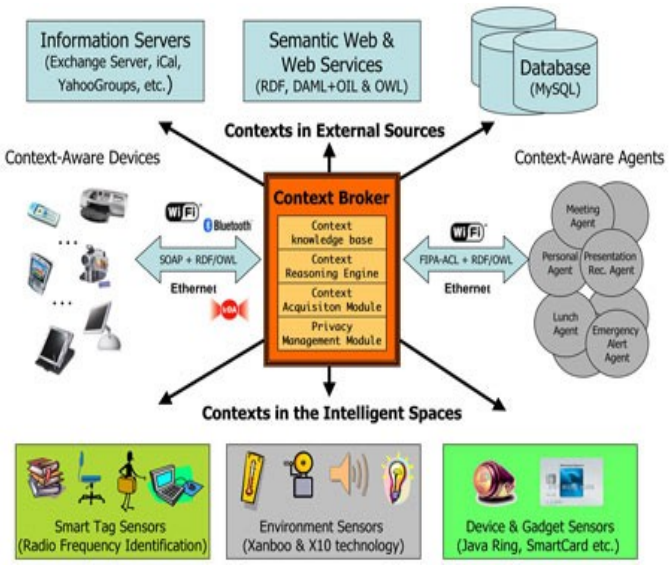



**Fig. 3.** Context Broker Architecture.

## 3. Context ontology

Context ontology is needed in order to deal with a heterogeneous character of context information. In particular, and ontology focuses on identifying objects by classifying them and characterizing them with properties [10]. Below we present an early attempt of a classification of context information, including identification of common properties and description of relationships between them. We also briefly introduce the Web Ontology Language [11], which can be used for representing context ontology.

### 3.1. Context taxonomy

We first aim fat establishing a simple taxonomy for context information, based on various approaches proposed in the literature.

In general, context information can consist of very different information. User's location is one of the most popular kinds of context information [12]. However, other common kinds of context include, e.g., user's identity, activity pattern, IP address and time of transaction.

We can identify the following four basic categories for context information [13], [14]:

**System context**: A mobile application has to take into account context information related to both the computing system it is running on, e.g. the particular type of mobile device, and to communication system being used, e.g. the particular type of wireless network. System context deals with any kind of context information related to a computing system, e.g. computer CPU, network, IP address, status of a workflow, wireless network, etc. In order to provide a fine-grained classification of system context, we can use model based on context information provided by different layers of Open System Interconnections (OSI) model [15]: application, presentation, session, transport, network, data link, and physical. Figure 4 provides an example of context information available per layer. For example, a workflow status would be provided by the application layer whereas an IP address would be related to the network layer.

**User context**: refers to any kind of context information related to the user and characterizing him. User context information can be user's age, location, medical history, etc. Biometric information, such as fingerprint, iris or face shape, is a part of user context, too. User context can be also related to his emotions [16], even if it is currently difficult to capture this kind of information in computing systems. User context can also include context information related to user's tasks, social connections, personal state, and spatio-temporal information [17]. The task context captures user's tasks, goals and current activities. The social context focuses on capturing the social relationships, such as family, friends, and colleagues. The personal context describes physical and mental information about the user. The spatio-temporal context refers to the location or movement of the user.

**Environmental context**: consist of any kind of context information related to the physical environment, which is not covered by system and user context. Environmental context information include, e. g., lighting, temperature, weather, etc.

**Temporal context**: defines any kind of context information related to time. Time and day are typical temporal context information.

Figure 4 depicts the taxonomy of context information described above.

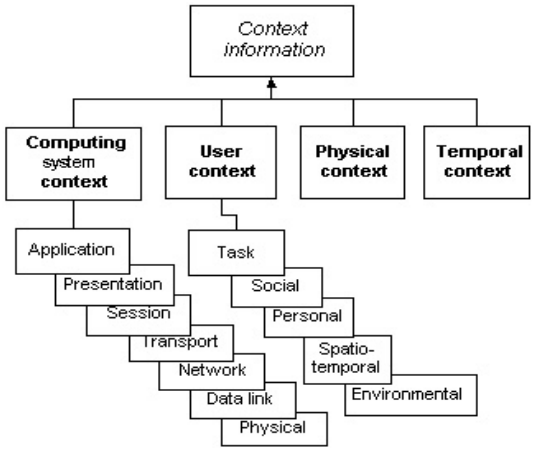



**Fig. 4.** Classification of context information.

Any context information can be classified as type-of object class as defined in Figure 4. For example, an IP address is considered as type-of Network context information which is a subtype-of System context. Our proposed taxonomy should be considered as a basis for context information classification, and therefore it can be extended. Nevertheless, classification of context information is not sufficient. It is necessary to identify the properties of this context information. In the next section, we describe a set of properties that we believe are common to many kinds of context information.

### 3.2. Attributes of context information

In this section our goal is to present a non-exhaustive list of attributes which characterize various kinds of context information. Figure 5 illustrates this basic set of attributes of context information. In particular, we have identified the three main properties of context information:

**Persistence of context information**: We consider persistence of context information as relative to the application life-time. In particular, static context information does not change (or changes very slowly). Static context information can include user's address, information name, room temperature, etc. At the contrary, dynamic context information changes much more often. Dynamic context information includes time, user's location, etc.

**Origin of context information**: context information can be provided by either an internal or an external source. For example, in the case of an application hosted by a mobile phone, context information coming from the mobile phone itself is considered as internal context information; whereas context information provided, e.g., by a GSM operator (such as user's location) is considered as external context information. We believe that the distinction between internal and external context information is important for the evaluation of, e.g., quality and trustworthiness of context information. In particular, internal context information, such as battery level, can be assumed more reliable than external context information such as network coverage guaranteed by the mobile operator.

**Quality of context information**: an important attribute of context information is its quality. For example, various approaches have been proposed in order to increase accuracy of GPS location information for outdoor user's localization or to provide user's location indoor based on Wi-Fi networks [18]. Gray [19] introduced criteria for evaluation of quality of the context information. Based on the quality of the context information, the acquisition can be repeated periodically, either in order to maintain the freshness of the context information or when some events in the system occur [7]. The importance of the quality will be further discussed in section IV. As stated previously, the above list of common property is not exhaustive. Nevertheless, we believe that it provides a good basis for further research.

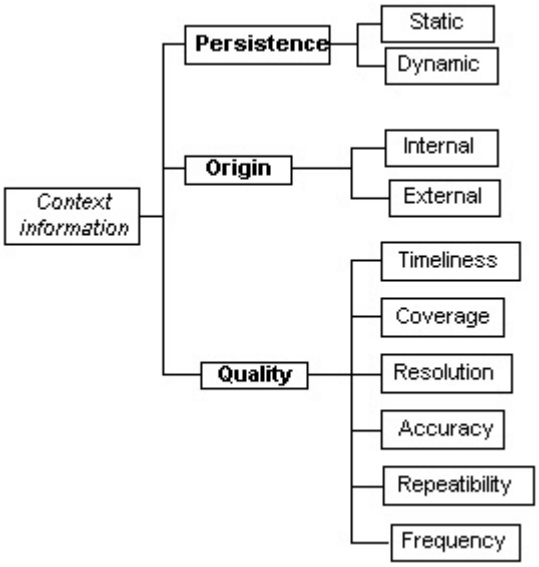



**Fig. 5.** Attributes of context information.

### 3.3. Context Information Relationship

Even if context information is clearly classified and if a list of common properties is established, relationships between different kinds context information have to be identified in order to enable further reasoning about context. For example, an e-Health application may acquire context information related to a patient's health status such as a patient's heart rate, pulse and blood pressure. We can assume that an equivalence relation exists between heart rate and pulse [11]. Moreover, we could state, e.g., the following correlation between context information: if pulse lower than 10 and blood pressure lower than 10 then health status is unconscious. During the reasoning phase, if pulse is not available, heart rate can be used for reasoning about the patient's health condition. Identification of relationships between context information permits aggregation of context information at the application level: instead of having different context information from different sources, correlation compiles sets of context information in single and comprehensive information, which can be used by the application. The goal of aggregation and reasoning about context information is to avoid flooding an application with too much context information and to provide a global context understanding. Nevertheless the quality of the reasoning about context information is based on the properties of the individual context information. Depending on the individual quality of a set of context information, reasoning about correlated imperfect context information determines the quality of this new context information [20]. The complexity of reasoning about imperfect information may constitute a significant obstacle for the success of context-aware system.

### 3.4. OWL-based Context Ontology

The Web Ontology Language (OWL) has been developed as a part of the Semantic Web initiative sponsored by World Wide Web Consortium (W3C) [11]. Based on the XML standard, OWL is a knowledge representation language for defining and instantiating of ontologies. Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) [21] is an ontology covering knowledge sharing, context information reasoning and inter-operability in ubiquitous environments. Expressed using OWL, SOUPA ontology consists of two sets of ontologies: SOUPA Core and SOUPA Extension. SOUPA Core defines a typical vocabulary for nine basic concepts: person, agent, belief-desire-intention, action, policy, time, space and event. For example, for the person concept, SOUPA Core defines attributes such as first name, last name or contacts. SOUPA Extension enables the definition of new set of vocabulary. Existing SOUPA extensions include location, image capture, and region connection calculus.

## 4. Context and security

Context ontology provides a conceptual mechanism for understanding of context at the application-level. This context can be further used at application-level for the adaptation of application's logic and management of security. However, such context-aware systems face important security challenges related to the use of context information: the main issue is to how to get secure and trusted context information. In section IV-A, we focus on the management of trusted and secure context information. The goal is to provide to context-aware system reliable context information. Afterwards, we discuss in section V how trusted context can be integrated as input for trust and security management in context-aware system. We also explain in that section the benefits, which mobile application can gain from using context-aware security.

### 4.1. Secure context

During each step of the life-cycle of context, see also Figure II-B, context-aware systems face specific security and trust challenges. We consider the three main security needs in computer system: confidentiality, integrity and availability [5].

- Confidentiality: focuses on protecting of computer system assets (Computer system assets include hardware, software, media storage and data) from unauthorized entities; For example, user can choose to disclose his age only to specific authorized entities.
- Integrity: focuses on ensuring that computer system assets can be modified only by authorized entities; For example, temperature measurements delivered by thermometer must be reliable and not modified by any entity. Context information spoofing is

- Availability: focuses on ensuring that computer assets must be available to the authorized entities. A GPS receiver must ensure to provide user's location anywhere and anytime.

Those three security goals are also relevant for a context-aware system, with respect to information exchanged between the context information provider and requestor during the context life-cycle. We have identified the following security issues related to exchange of context information:

**Confidentiality of context information**: (also called context information privacy). Considering user's context information (see also section III-A), user should be able to protect personal information such as his health status, or medical history. Jason proposes an architecture so-called Confab in order to provide privacy in ubiquitous computing [22]. Confab framework has been designed for protecting a user's location information in ubiquitous systems. It is based on an analysis of privacy needs for end-users and application developers. The main idea is that personal information are captured, stored and processed as much as possible on the user's device. Users can apply security control on their choice on those data. Context views have been introduced by Shankar [23]. Context view is a fraction of an entity's context that is relevant to the application. In order to protect context information confidentiality, delivered context information can be controlled in context view. Bussard et al proposed security mechanisms for protecting privacy of context information. [24] These context information are used for further trust relationship establishment between two entities.

**Integrity of context information**: focuses on guaranteeing that the provided context information has not been corrupted by a third party. Hash functions or public key digital signatures can provide context integrity.

- Hash function: We consider two kind of hash function: un-keyed and keyed hash functions. In the first case, un-keyed hash functions, such as MD5 or SHA-1, provide a very low level of data integrity with respect to the use of context in application adaptation. With keyed hash function, such as MAC, we have the issue of the establishment of pre-shared key between the context information providers and the context-aware systems.
- Public key digital signature: the PKI-based approach might not be always suitable for context-aware systems, especially in distributed systems including low-cost sensors [25].

**Availability of context information**: denial of service attacks could be easy to perform on a context information provider. Possible denial-of-service attacks on context information providers include the sleep deprivation torture or CPU exhaustion [25]. However, we do not focus on this security challenges here.

When considering that the application logic and the security configuration can depend on context, we have to ensure that context information is trustworthy. In particular, we have to be able to estimate the level of trust a context information requester can put in the delivered context information. Due to the fact that context information provider and the context-aware system might be members of different administrative domains, it is very important to be able to differentiate trustworthy parties from un-trustworthy ones. Defining trust mechanisms and metrics is not an easy task [26], [27]. In the scope of context-aware system, we consider that trust refers to the reliability and accuracy of context information. In this sense, trust is related to the quality of the delivered context information, but not restricted to it. We can evaluate the trust about delivered context information by computing the distance between it and the real context. The following approaches can be used in order to compute this distance and evaluate the trustworthiness of delivered context information:

**Statistical analysis of context information**: The idea is to perform statistical analysis of the value of context information in order to detect unreliable ones. A simple example is to take an average of temperature values provided by different thermometer in the same room. Such naïve approach raises the following issue: if the temperature values collected are (10; 10; 11; 50) we get an average of over 20. It is obvious that the last value is a wrong one, and the temperature in the room should be 10. A simple solution is to use median that detects that value 50 is out of the scope, and eliminate the context information provider delivering this temperature value as an unreliable provider. Of course, real-life implementation requires use of much more sophisticated statistical tools.

**Distributed reputation network**: Trustworthiness is often described as the expectation of cooperative behavior [28]. Usually the trustworthiness is based on previous experience with the same party. The problem is that often there has not been any direct interaction before. In this case, an entity can establish trust in its communication partner based on the latter's reputation [29-31]. Reputation is based on the collection of evidence of good and bad behavior. In the case where we use sensors in order to identify and authenticate users, their reputation depend, e.g., on whether the sensors input led to correct authentication or to a violation of a security policy. Reliability of context information is computed based on the previous experiences with a context information provider. The mathematical foundations for reputation management are rooted in statistics and probability [32]. Trust context views [33] have been introduced by Robinson in order to establish a degree and state of trust within a certain interactive context. This approach doesn't consider the authentication of context information provider.

**Confidence value**: In the scope of Gaia architecture, Al-Muthadi et al [34] define a notion of confidence values in context-aware authentication. Gaia architecture establishes a confidence value based on the authentication devices and protocol used. For example, iris recognition would have a better confidence value

than a fingerprint authentication device. The use of challenge-response protocol would be considered with a strong confidence value than a user's identity sent in clear text.

Above considerations about security and trust are relevant for discovery and acquisition of context information. Reasoning about context information raises another security challenge. Indeed, reasoning about context information often deals with integrating several pieces of context information into another piece of context information. For example, from a patient's heart rate and blood pressure, we can evaluate his health condition. Assuming that patient's heart rate and blood pressure has different level of trust and security, the question is about the level of trust and security about the patient's health condition.

## 4.2. Context-aware security

Context information can be also used in order to reconfigure and enhance security of the system and of the applications. We define context-aware security as dynamic adaptation of system security policy according to the context. For example, a context-aware system can exploit user's location for access control: if the user is in a public area, such as airport, he should not be able to display confidential data in a way it could be seen by a third party. Context information could also be used in order to automatically reconfigure security mechanisms in order to provide a predefined level of security, at the same time optimising use of resources. For example, email messages send by a mobile worker using WLAN hotspot as an access point for his PDA can be automatically encrypted, whereas the same massages could be sent in plaintext when connected to an access point belonging, which employs the worker.

As we have already mentioned, one of the main challenge in context-aware security is estimation of trust we put in context information. This is especially true, if context information is to be used in order to enforce security policies in mobile applications or to alter security configuration of a portable device.

Below, we describe trust management and access control as two candidates for security services integrating context information.

### 4.3. Access control

Access control is a set of mechanisms and processes that aim at protecting assets of a computer system from being accessed by unauthorized entities. Access control determines whether an access request to a resource or data is granted or denied [35]. Access control policy defines the set of controls or permissions in the computing system. Permission is normally associated with a subject, such as a user. It is a set of rights for a specific object. Access control policy is commonly enforced through identification of a subject and a decision of granting or denying access to an object, according to the permissions associated to the subject identity. For example, user A authenticates himself to a computing system. A requests access to file C. As permission B allows user A to read and write in file C, access control enforcement grants access to file C to user A. Access control targets the three main security goals: authentication, authorization and accounting. Authentication is the process whereby one party is assured of the identity of a second party involved in a protocol, and that the second has actually participated (i.e., is active at, or immediately prior to, the time the evidence is acquired). Authorization deals with the rights on an asset associated to a principal. Finally, accounting focuses on logging all requests for access, as well as the corresponding access decisions. Context-aware access control can be easily illustrated by the following example: a doctor should have the rights to book a room for a patient in a hospital or plan for a surgery only if he is physically in the hospital and not on vacation. In our e-Health scenario, emergency team member can get access to all health information about victim, only if the latter is unconscious. Based on context information, proximity-based and encounter-based access control can be defined [36]. Proximity-based access control defines a set of security rules based on the proximity of an entity to other entity or group of entities. Another example of a context-aware system that exploit user's location for access control could be as follow: if the user is in a public area, such as an airport terminal, he should not be able to display confidential data in a way it could be seen by untrusted third parties. In the scope of access control, we can also consider context-aware delegation of rights. For example, if a manager is out of office, he can delegate some of his rights to his secretary until he is back in his office. Depending on the urgency of certain task, context-aware system can decide whether delegation can take place or not with respect to particular tasks.

Several architectures have been developed for context-aware access control:

**Environmental Roles**: Covington et al [37] propose a uniform access control framework for environmental roles. It is an extension to the Role-Based Access Control (RBAC) model. In RBAC model, permissions are associated with roles. In an administrative domain, a role can be a developer or a manager. A role determines

the user's position or ability in an administrative domain. Likely, an environmental role is a role that captures environmental conditions. Environmental roles are based on General Role Based Access Control [38]. Unlikely in the RBAC model which is only subject-oriented, GRBAC allows defining access control policy based on subject, object or environment. An implementation of the environmental roles is provided by, e.g., CASA.

**Gaia**: Roman et al [39] defined generic context-based software architecture for physical spaces, so-called Gaia. A physical space is a geographic region with limited and well defined boundaries, containing physical objects, heterogeneous networked devices, and users performing a range of activities. Derived from the physical space, Active Space provides to the user a computing representation of physical space. Active Space helps the user to interact with the physical space. Cerberus is a framework for context-aware identification, authentication and access control and reasoning about context, based on Kerberos [40] authentication and Gaia [34]. Cerberus focuses on user's identification via user's context information such as fingerprint, voice and face recognition.

**Authorization mechanisms for Intranet**: This context-aware authorization architecture, based on Kerberos authentication, enables to activate or deactivate role assigned to a user depending on his context [41]. For example, if a user is in a un-secure place such as airport terminal, the access to sensitive data is denied whereas in the corporate building of the user, he has access to the confidential data.

### 4.5. Trust management

Several trust models have been proposed in the literature [26]. In the scope of ubiquitous environment, we target the trust management of mobile application across multiple domains. It means that mobile application involved in collaboration may not trust a priori each other. When considering collaborative mobile applications, we can identify the following history information, which can influence the trust relationship during the collaboration:

**Respect of common security policy**: Assuming that mobile application negotiate a common security policy at the beginning of the collaboration, if one of the entity involved in the collaboration breaks the common security policy it should decrease the trust that the others entities will have in the latter.

**Respect of the common goal**: The collaboration aims at achieving a specific goal, e.g., sharing a file in a secure manner, exchange contact, etc. If actions of one the entities is not focused on the goal of the collaboration, it level of trust should decreases. At the contrary, if an entity fulfills its entire obligation in the collaboration, its level of trust should increase.

As we discussed in Section IV-A, trust is a complex concept. Defining a generic model of trust is a very challenging and so far unsolved issue [27]. By exploiting the context information, we aim at managing trust between entities in collaboration on a case by case basis. We also aim at providing a dynamic evaluation of level of trust in context information itself. This issue is directly related to risk management. In particular, mobile applications can accept the risk of using un-trustable context information, either if it is in urge or it doesn't really care of the reliability of the context information. On the other hand, user or application can decide to fix a threshold of trust, below which context information is not taken into account when considering adaptation of application logic or security mechanisms.

## 4.6. Ongoing and future work

The secure context awareness and context aware security are currently field of active research in both academic and industrial community. Mobile Workers' Secure Mobile Application Collaboration in Ubiquitous Environment (MOSQUITO) is a joint European research project funded under EU FP6 IST Programme [21]. It aims at providing a secure framework for collaborative mobile business application. In particular, within MOSQUITO we aim at developing a framework for context-aware security services in ubiquitous computing environment. In order to deal with security of context information, we introduce in MOSQUITO a concept of Context Information Provider (CIP). Trusted CIP (TCIP) is responsible for providing accurate and secure context information to context-aware applications. We consider several classes of TCIPs:

**Third Party TCIP**: a stand-alone system operated by a party belonging to a different administrative domain than the entity requesting context information. This one is for instance operated by a mobile network operator.

**Intra-Domain TCIP**: a service that is operated by an entity belonging to the same administrative domain as the entity requesting context information. Intra-domain TCIP can be a stand-alone service or a service integrated in another server.

**Embedded TCIP**: a tamper-resistant component of a (mobile) device that ensures integrity and authenticity of provided context information.

## 5. Conclusions

In this paper we have focused on investigation of two different aspects of security related to context information.

First, we presented challenges related to security of context information. Security challenges in context-aware systems include integrity, confidentiality and availability of context information, as well as end user's privacy. Another important issue is trustworthiness of context information, i. e. the amount of trust, which a context information requester can put in the delivered context information.

Second, we have discussed how availability of additional context information could be used in order to manage and optimize security configuration of the mobile devices and services offered to the user.

## 6. Disclaimer

*IST-Directorate General / Integrating and strengthening the ERA*: the project MOSQUITO is supported by the European Community. This document does not represent the opinion of the European Community. It is also the sole responsibility of the author and not the responsibility of the European Community using any data that might appear therein.

## References

1. F. Stajano, *Security for Ubiquitous Computing*, John Wiley and Sons, Feb. 2002.
2. A. K. Dey and G. D. Abowd, *Towards a better understanding of context and context-awareness*, in Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, Apr. 2000. ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf
3. *Toward a better understanding of context attributes*, in Proceedings of the second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMMW 04). IEEE Computer Society, 2004.
4. A. K. Dey, *Understanding and using context*, Personal and Ubiquitous Computing Journal, vol. **5** (1), pp. 4–7, 2001. http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf
5. C. P. Pfleeger, *Security in Computing*, Prentice-Hall, Inc., 1997.
6. R. J. Orr and G. D. Abowd, *The Smart Floor: A Mechanism for Natural User Identification and Tracking*, in Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands, April 2000, pp. 1–6.
7. N. J. Siljee B. I. J, Bosloper I. E, *A classification framework for storage and retreival of context*, in Proceedings of First International Workshop on Modeling and Retrieval of Context (MRC2004), 2004.
8. A. K. Dey, *Providing architectural support for building context-aware applications*, Ph.D. dissertation, Georgia Institute of Technology, Nov. 2000. http://www.cc.gatech.edu/fce/ctk/pubs/deythesis.pdf
9. H. L. Chen, *An intelligent broker architecture for pervasive contextaware systems*, 2004.
10. *What are ontologies, and why do we need them?* 1999.
11. W3C, *Owl - web ontology language*, http://www.w3.org/2004/OWL/
12. U. Hengartner and P. Steenkiste, *Implementing access control to people location information*, in SACMAT '04: Proceedings of the 9th ACM symp. on Access control models and technologies. ACM Press, 2004, pp. 11–20.
13. B. Schilit, N. Adams, and R. Want, *Context-aware computer applications*, Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85–90, Dec. 1994.
14. G. Chen and D. Kotz, *A survey of context-aware mobile computing research*, Tech. Rep. Dartmouth Computer Science Technical Report TR2000-381, 2000.
15. ISO, *Osi - open system interconnection.* http://www.iso.org/
16. R. W. Picard, *Affective computing: challenges*, Int. J. Hum.-Comput. Stud., vol. 59, no. 1-2, pp. 55–64, 2003.
17. A. K.-P. Marius Mikalsen, *Representing and reasoning about context in a mobile environment*, 2004.
18. J. Small, A. Smailagic, and D. Siewiorek, *Determining user location for context-aware computing through the use of a wireless lan infrastructure*, Carnegie-Mellon University, Tech. Rep. Project Aura report. http://www.cs.cmu.edu/ aura/docdir/small00.pdf
19. P. Gray and D. Salber, *Modelling and using sensed context information in the design of interactive applications*, in Engineering for Human-Computer Interaction: 8th IFIP International Conference, EHCI 2001, (LNCS), vol. 2254, 2001. http://www.springerlink.com
20. K. Henricksen and J. Indulska., *Modelling and using imperfect context information*, in In Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
21. H. C. F. P. T. Finin and A. Joshi, *Soupa: Standard ontology for ubiquitous and pervasive applications*, 2004. http://ebiquity.umbc.edu/v2.1/get/a/publication/105.pdf

22. J. I. Hong and J. A. Landay, *An architecture for privacy-sensitive ubiquitous computing*, in MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services. ACM Press, 2004, pp. 177–189.
23. N. Shankar and D. Bafanz, *Enabling secure ad-hoc communication using context-aware security services*, in UNBICOMP 02: Workshop on Security in Ubiquitous Computing, 2002. http://www.teco.edu/philip/ubicomp2002ws/organize/palo.pdf
24. M. R. Bussard L., Roudier Y., *Untraceable secret credentials: Trust establishment with privacy*, in PERCOMMW'04. Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
25. F. Stajano and R. Anderson, *The Resurrecting Duckling: Security issues for ad-hoc wireless networks*, in Proceedings of the 7th International Workshop on Security Protocols, ser. Lecture Notes in Computer Science, M. R. B. Crispo, Ed. Springer, 1999.
26. K. Wrona, *Cooperative communication systems*, Ph.D. dissertation, Department of Wireless Communications, University of Aachen, Germany, 2005.
27. M. A. P. S. T. M. C. Michael, *Clifford Information Security Analyst*,The Aerospace Corporation Matt Bishop Associate Professor The University of California at Davis "Defining, computing and interpreting trust," 2000. http://csdl.computer.org/comp/proceedings/acsac/2000/0859/00/08590088.pdf
28. E. I. Covington M. Ahamd M. and V. H., *Parameterized authentication*, 2004. http://www.springerlink.com
29. S. Ganeriwal and M. B. Srivastava, *Reputation-based framework for high integrity sensor networks*, in SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. ACM Press, 2004, pp. 66–77.
30. B. Yu and M. P. Singh, *An evidential model of distributed reputation management*, in AAMAS '02. Bologna, Italy: First International Conference on Autonomous Agents and MAS, July 2002.
31. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, *The eigentrust algorithm for reputation management in p2p networks*, in In Proceedings of the Twelfth International World Wide Web Conference, 2003.
32. G. Shafer, *A mathematical theory of evidence*, 1976.
33. P. Robinson, *Trust context spaces an infrastructure for pervasive security in context-aware environments*, 2003.
34. R. C. Jalal Al-Muhtadi Anand Ranganathan and M. D. Mickunas, *Cerberus: A context-aware security scheme for smart spaces*. 2003. http://gaia.cs.uiuc.edu/papers/cerberus.pdf
35. P. Samarati and S. de Capitani di Vimercati, *Access control: Policies, models, and mechanisms*, 2001.
36. R. S. Roshan K.T, *Models, protocols and architecture for secure pervasive computing: Challenges and research directions*, in PERCOMMW' 04. Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
37. Z. Z. Michael J. Covington Prahlad Fogla and M. Ahamad, *A contextaware security architecture for emerging applications*, in Proceedings of the Annual Computer Security Applications Conference (ACSAC), Las Vegas Nevada USA, 2002.
38. M. J. C. M. J. Moyer and M. Ahamad., *Generalized role-based access control for securing future applications*, 2000.
39. M. R. C. K. H. R. C. A. R. R. H. Campbell and K. Nahrstedt, *Gaia: A middleware infrastructure to enable active-spaces*, 2002. http://gaia.cs.uiuc.edu/papers/GaiaSubmitted3.pdf
40. B. C. Neuman and T. Ts'o., *Kerberos: An authentication service for computer networks*, 1994. http://nii.isi.edu/publications/kerberos-neuman-tso.html
41. M. L. Wullems Chris and A. Clark, *Toward context-aware security: an authorization architecture for intranet environments*, 2004.

XXI Autumn Meeting of Polish Information Processing Society
Conference Proceedings, pp .267-272
ISBN 83-922646-0-6


# Multimedia in management facing globalization processes and cognitivistics

Jerzy Wąchol[1]

[1] Faculty of Management, AGH-University of Science and Technology,
Gramatyka 10, 30-067 Kraków, hol@uci.agh.edu.pl
www.zarz.agh.edu.pl

**Abstract.** The paper covers the use of multimedia and the Internet in management, concentrating mainly on globalization and management process (planning, organization, motivation and control). The use of modern multimedia and the Internet in management must be thoroughly thought over because it has both chances and new possibilities and dangers for companies, employees and clients in the new age of knowledge, information technologies and globalization. The paper also presents statistical data in selected countries about the saturation of multimedia in the World.

## 1 Introduction

Thanks to the significant technological process of last decades the use of multimedia in modern world is really enormous and it still increases. Computers, computer networks, mobile phones, calculation centers and data bases, automatic information and management centers, mass media, radio or television are already a constant element of management practice.

Multimedia are practically at every management step, planning, organization, motivation and control– especially at operational process. Moreover, they are widely applied to financial operations, banks, information (also secret) transfer in marketing, public relations, etc.

Thanks to mass popularization of multimedia the world becomes a global village. All information, ideas or tasks can be very quickly transferred and spread to every part of our planet. Moreover, it gives the possibility of service and technology transfer on a large scale in globalizing economy. All this, as any great invention, has its possibilities and chances but, on the other hand, can cause enormous dangers. Technologies which are imperfect and are also badly used can be the reason of many problems and tragedies. Legal system which, as usual, does not follow technological changes and global economy, is also important here.

Due to its limitations, the paper contains only selected aspects of management, globalization and net economy processes, presents the popularization of multimedia in the world quoting statistic data related to Gross Domestic Product and demographic data. This paper contains also elements of cognitivisitcs and influence of multimedia on man and their activity.

## 2 Globalization process and multimedia

In modern world there are many voices both for globalization and against it, from extreme liberalism to extreme etatyzm. It seems , however, that for the civilization point of view the globalization process cannot be avoided. There is still a question about the shape of globalization, its implementation, its beneficiaries, etc. There is probably a need of a compromise between extreme ideologies and interests of individual groups.

Tendencies of constant fight of different interests and rapid appearances of new economic phenomena on one hand are limited by the tendencies of closing these processes within clear legal state regulations connected with the informative revolution. These processes acquired a new character especially at the turn of 20 and 21 centuries. From these points of view, computer networks can become the basic element of the future global management system which will cover the whole human life.

Globalization processes have taken place at different scale and pace for hundreds of years. But they acquired a decisive dynamics at the time of the great geographical discoveries. They accelerated fast with consecutive,

revolutionary scientific-technical discoveries, from steamships and railways up to modern jet planes, large tankers, through telegraph and telephone up to modern Internet [1].

The last 20th century, know as the age of steam and electricity played a very special role, the present 21st age is supposed to be the age of information and knowledge. In the present world many opinions both and against globalization can be encountered, from extreme neo-liberalism to extreme state control. It seems, however, that from the civilization point of view, globalization processes are not to be avoided. Here comes the question: what kind of globalization will it be, how will it be implemented, who will it suit, whose interests will be represented, etc. Probably, a certain compromise between these ideologically extreme views and interests of individual groups is necessary. Tendencies to constant confrontation of different interests and spontaneous formation of new economic phenomena on the one hand, are limited by the tendencies to keep these processes within clear, legal regulations connected with the information revolution. These processes developed a particular character especially at the end of the 20th century and at the beginning of the 21st century. From this point of view, computer networks can become a basic element of future global management system, covering the whole human life.

Conditions of the present globalization process create, of course, many old and new problems concerning, among others, social, economical, technical, technological, legal, organization, cultural, awareness and ecological areas. Polarization and differentiation of the world, e.g. into richer and poorer is growing, migrations, capital and culture flows are becoming more intensive. The control of the state over multinationals is usually limited. Supranational computerization and communication standards structure is being built. It creates both opportunities and threats form the point of view of states, nations, business groups and ordinary people. The question arises: how will the globalization process be conducted and who will benefit from it?

The use of e-business, of course, brings many new possibilities or changes in economy and management, but requires modern, technological infrastructure – this requirement is connected with large, financial investments. As it can be seen from the data presented, e-business develops fastest in rich areas, such as the USA and EU, where about 2/3 of the world's GDP is produced, causing and additional large economic growth using multimedia (like the loop of positive feedback) in spite of the fact that in rich, developed countries the growth dynamics of GDP is very small (0.1-3%) per year. However, non-material values, like in a typical information society, begin to dominate. Although USA and EU goals are similar (information society) the way to their achievement is different. The Americans believe more in market self-regulation and business profits, the EU also stresses state regulations and social values – hence the program e-Europe. Probably, a certain compromise to achieve mutual goals (within the frames of globalization) and fulfill the rowing needs of the population can be made.

Table 1 shows the saturation of multimedia in different world countries in relation to number of population and GDP per head. It can be stated that the bigger the saturation of multimedia, the higher living standard is.

## 3 The use of multimedia in management

The use of multimedia, microprocessors, computers and computer networks is common today – however, I think that it has a unique and large significance in management because it makes management more effective and optimum, where decisions can be taken with the use of modern forecasting methods, data processing, simulation calculations, etc. It does not relieve the manager from decision-making processes, but it can help him to make the right decisions. The planning function can be supported with data processing, multiple calculations and simulations. The organization and motivation function can be reinforced with data based, modern communication, calculation, multimedia training means.

The control function can be largely used through reports, monitoring, saving nearly everything including the place of an employee on the Earth in a given moment with the use of GSM/DCS and UMTS telephones and GPS satellite positioning equipment. Management Process [2] presenting the regulation equipment system is shown in Fig.1 and the operational system of the enterprise in Fig. 2.

**Table 1.** Saturation of multimedia in relation to GDP and number of population in selected World countries

| | State /year | Inter--net users mln | Popula--tion mln | GDP real per capita thousands USD | GDP real growth rate % | Inter--net hosts mln | Telephones main lines in use mln | Telephones mobile cellular mln |
|---|---|---|---|---|---|---|---|---|
| | | ↓ /year | 2004 | 2003 | 2003 | /year | /year | /year |
| 1 | USA | **159,3** /03 | 293,0 | 37,8 | 3,1 | 115,3/02 | 181,6/03 | 158,7/03 |
| 2 | China | **79,5** /03 | 1298,8 | 5,0 | 9,1 | 0,16 /03 | 263,0/04 | 269,0/03 |
| 3 | Japan | **57,2** /03 | 127,3 | 28,2 | 2,7 | 12,9 /03 | 78,1 /02 | 86,6/ 03 |
| 4 | Germany | **39,0** /03 | 82,4 | 27,6 | -0,1 | 2,7 /04 | 54,3 /03 | 64,8 /03 |
| 5 | Korea S. | **29,2** /03 | 48,5 | 17,0 | 3,1 | 0,7 /01 | 22,9 /03 | 33,6 /03 |
| 6 | G. Britain | **25,0** /02 | 60,3 | 27,7 | 2,2 | 3,4 /04 | 34,9 /02 | 49,6 /02 |
| 7 | France | **21,9** /03 | 60,4 | 27,6 | 0,5 | 2,3 /04 | 33,9 /03 | 41,6 /03 |
| 8 | Italy | **18,5** /03 | 58,0 | 26,6 | 0,4 | 1,4 /04 | 26,6 /03 | 55,9 /03 |
| 9 | India | **18,5** /03 | 1065,1 | 2,9 | 8,3 | 0,09 /03 | 48,9 /03 | 26,1 /03 |
| 10 | Canada | **16,1** /03 | 32,5 | 29,8 | 1,7 | 3,2 /03 | 19,9 /03 | 13,2 /03 |
| 11 | Brazil | **14,3** /02 | 184,1 | 7,6 | -0,2 | 3,1 /03 | 38,8 /02 | 46,3 /03 |
| 12 | Mexico | **10,0** /02 | 104,9 | 9,0 | 1,3 | 1,3 /03 | 15,9 /03 | 28,1 /03 |
| 13 | Spain | **9,7** /02 | 40,3 | 22,0 | 2,4 | 1,1 /04 | 17,6 /03 | 37,5 /03 |
| 14 | Australia | **9,5** /02 | 19,9 | 29,0 | 3,0 | 2,8 /03 | 10,9 /03 | 14,3 /03 |
| 15 | Poland | **9,0** /03 | 38,6 | 11,1 | 3,7 | 0,8 /04 | 12,3 /03 | 17,4 /03 |
| 16 | Netherlands | **8,5** /03 | 16,3 | 28,6 | -0,7 | 4,5 /04 | 10,0 /02 | 12,5 /03 |
| 17 | Russia | **6,0** /02 | 143,7 | 8,9 | 7,3 | 0,5 /04 | 35,5 /02 | 17,6 /02 |
| 18 | Turkey | **5,5** /03 | 68,9 | 6,7 | 5,8 | 0,4 /04 | 18,9 /03 | 27,8 /03 |
| 19 | Sweden | **5,1** /02 | 8,9 | 26,8 | 1,8 | 0,9/ 04 | 6,6 /02 | 7,9 /02 |
| 20 | Iran | **4,3** /03 | 69,1 | 7,0 | 6,1 | 0,005/04 | 14,5 /03 | 3,4 /03 |
| 21 | Argentina | **4,1** /02 | 39,1 | 11,2 | 8,7 | 0,7 /03 | 8,0 /03 | 1,5 /02 |
| 22 | Romania | **4,0** /03 | 22,3 | 7,0 | 4,9 | 0,05 /04 | 4,3 /03 | 6,9 /03 |
| 23 | Austria | **3,7** /03 | 8,1 | 30,0 | 0,7 | 0,4 /04 | 3,9 /03 | 7,0 /03 |
| 24 | Portugal | **3,6** /02 | 10,5 | 18,0 | -1,3 | 0,3 /04 | 4,3 /03 | 9,3 /03 |
| 25 | Belgium | **3,4** /03 | 10,3 | 29,1 | 1,1 | 0,2 /04 | 5,1 /02 | 8,1 /02 |
| 26 | Hong-Kong | **3,2** /03 | 6,9 | 28,8 | 3,3 | 0,6 /03 | 3,8 /03 | 7,2 /03 |
| 27 | Africa S. | **3,1** /02 | 42,7 | 10,7 | 1,9 | 0,3 /03 | 4,8 /02 | 16,9 /03 |
| 28 | Denmark | **2,7** /03 | 5,5 | 31,1 | 0,0 | 1,2 /04 | 3,6 /03 | 4,7 /03 |
| 29 | Czech Rep. | **2,7** /03 | 10,2 | 15,7 | 2,9 | 0,3 /04 | 3,6 /03 | 9,7 /03 |
| 30 | Finland | **2,7** /02 | 5,2 | 27,4 | 1,9 | 1,2 /04 | 2,5 /03 | 4,7 /03 |
| 31 | Switzerland | **2,5** /03 | 7,5 | 32,7 | -0,5 | 0,7 /04 | 5,5 /02 | 6,1 /03 |
| 32 | Norway | **2,3** /02 | 4,6 | 37,8 | 0,6 | 0,6 /04 | 3,4 /02 | 4,2 /03 |
| 33 | Israel | **2,0** /02 | 6,2 | 19,8 | 1,3 | 0,5 /04 | 3,0 /02 | 6,3 /02 |
| 34 | Greece | **1,7** /03 | 10,6 | 20,0 | 4,7 | 0,2 /04 | 5,2 /03 | 8,9 /03 |
| 35 | Hungary | **1,6** /02 | 10,0 | 13,9 | 2,9 | 0,4 /04 | 3,6 /02 | 6,9 /02 |
| 36 | Slovakia | **1,4** /03 | 5,4 | 13,3 | 3,9 | 0,09 /04 | 1,3 /03 | 3,6 /03 |
| 37 | Belarus | **1,3** /03 | 10,3 | 6,1 | 6,8 | 0,005/04 | 3,0 /03 | 1,1 /03 |
| 38 | Ireland | **1,2** /03 | 3,9 | 29,6 | 1,4 | 0,2 /03 | 1,9 /03 | 3,4 /03 |
| 39 | Latvia | **0,9** /03 | 2,3 | 10,2 | 7,4 | 0,05 /04 | 0,6 /03 | 1,2 /03 |
| 40 | Ukrainian | **0,9** /03 | 47,7 | 5,4 | 9,4 | 0,1 /03 | 10,8 /02 | 4,2 /02 |
| 41 | Lithuania | **0,7** /03 | 3,6 | 11,4 | 9,0 | 0,06 /04 | 0,8 /03 | 2,1 /03 |
| 42 | Slovenia | **0,7** /02 | 2,0 | 19,0 | 2,3 | 0,04 /04 | 0,8 /03 | 1,7 /03 |
| 43 | Uruguay | **0,4** /0,2 | 3,4 | 12,8 | 2,5 | 0,09 /03 | 0,9 /02 | 0,7 /02 |
| 44 | Zimbabwe | **0,5** /02 | 12,6 | 1,9 | -13,6 | 0,004/03 | 0,3 /03 | 0,4 /03 |
| 45 | Kenya | **0,4** /02 | 32,0 | 1,0 | 1,5 | 0,008/03 | 0,3 /03 | 1,5 /03 |
| 46 | Luxemb. | **0,2** /03 | 0,5 | 55,1 | 1,2 | 0,03 /03 | 0,4 /02 | 0,5 /02 |
| 47 | Cuba | **0,1** /02 | 11,3 | 2,9 | 2,6 | 0,002/03 | 0,6 /02 | 0,02 /02 |
| 48 | Zambia | **0,07** /03 | 10,5 | 0,8 | 4,0 | 0,002/03 | 0,09 /03 | 0,2 /03 |
| 49 | World | **604,2** /03 | 6379,2 | - | 3,8 | - | 843,9 | - |

Source: Own elaborate on the basis of [6]

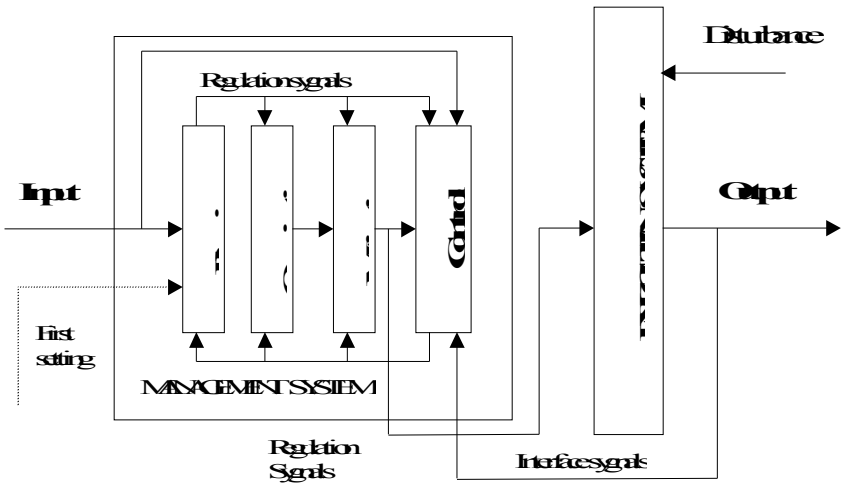



**Fig. 1.** Management Process of the regulation equipment system
Source: [3]

The use of multimedia, computers and the Internet considering management functions is shown in Table 2.

**Table 2.** The use of multimedia, computers and Internet in management

| MANAGEMENT FUNCTIONS | | |
|---|---|---|
| **Planning** | **Organization and motivation** | **Control** |
| – forecasting<br>– simulation calculation<br>–data processing, modelling, etc. | – data bases<br>– communications and identification means<br>– saving<br>– staff training and teaching, etc. | – monitoring<br>– reports,<br>– calculations,<br>– comparisons<br>– positioning, etc. |

Source: own research

Although it seems obvious that the use of multimedia, computers, the Internet, GSM/DCS/UMTS phone systems, etc. may only help in management and economy it may also happen that the wrong use of these techniques can lead to failure or even a tragedy. The manager who believes in the only right computer prints and makes his decisions on their base can fail since computer technology, though more and more sophisticated, is still very limited, based on the already existing programs which often do not take into consideration all the assumptions. In practice the manager's „thinking" cannot be substituted by computers in management process, it can however help him significantly.

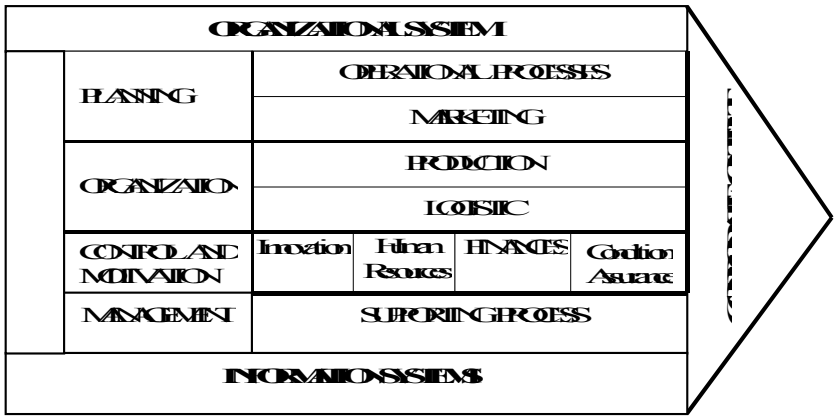



**Fig. 2.** The operational system of the enterprise.
Source: own elaborate on the basic of [4]

Relation between power, money and information knowledge is shown in Fig. 3.

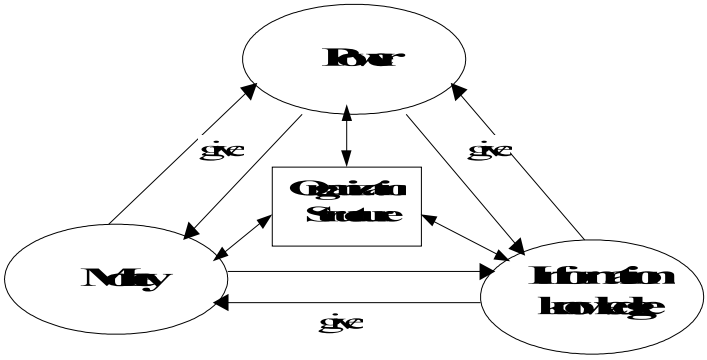



**Fig. 3.** Relation between power, money and information knowledge
Source: own research.

The next problem is connected with the implementation of modern computer and communication systems in companies. I think that many mistakes may be made here. On the other hand, a too comprehensive manager's control including the positioning of an employee at a given moment – can lead to dissatisfaction of the employees who are not used to such methods, their stress and, consequently, worse work results. In case when the employees are against it, are not mentally prepared because of e.g. psychological reasons and safety, I think that it is not worth using it, the employee can be evaluated by the work they have done without an expensive observing their movements and position. The use of new technologies and new management methods by force at an area and with people who are totally unprepared for this – can bring about most fatal, expensive and thoughtless results. The theory of management says that an employee needs some minimum of freedom in order to perform their duties well.

Another danger of thoughtless introduction of modern or pseudo-modern computer technologies can be massive losses of states, firms and clients themselves due to viruses and badly protected computer telecommunication, cheque transfers and secret information networks, etc. Here, great care and mastering of protection systems using microprocessor with an algorithm credit and phone cards – and not ordinary, magnetic ones, must be used.

## 4 Cognitivistics, medial experience and human behavior

New information and informative technologies era makes the surrounding world a world of collective imagination. This concerns mostly the use of mass media such as radio, television or computer networks which, to some extent, can create a virtual reality in human minds and psychology.

Even if some facts have never existed or their influence was relatively small, thanks to mass media they can exist medially. Then they start existing in human psychology, thus creating specific experiences and changes in human behavior, also creating new, but not necessary useful knowledge. People should approach the transferred and created information with care, criticism and intelligence and not consider such information as being true–if one does not experience it consciously and physically. Then, his world may become virtual and separated form reality. There must be a filter in human minds which separates information more or less important, true or false, etc.

The development of multimedia technologies aims probably at taking over the functions of television, radio, VCR, press, books, telephone, mail, fax, library, data bases or game machines by computers, computer programs and computer networks. In future it may only be one computer machine with attached interfaces. When people's mentality changes and adequate portable screens are applied even books will be better read from computer screens since there will be additional possibilities, e.g. fast finding of phrases, putting thousands of books on one mini-disc, possibility of marking fragments of texts for better remembering and making interactive versions of drawings and pictures. It may be difficult and expensive and require new technologies but, at the same time, there appear new possibilities for publishers and writers who must learn how to use these new methods of transfer and must know how to write a book or an article in the electronic version. Although new technologies are a challenge to mankind and give a wider scale of possibilities, they are still at the level of tools (programmed, shaped) which are used by a more complicated and intelligent creature–man.

There is no doubt that the computer having specific properties will be a kind of a hyper-medium and will make an intellectual aid in processing information by man. The analysis of this problem is not easy because it needs analyzing many areas which are deeply rooted in many sciences known under a mutual name of cogniti-

vistics [5]. Due to the limits of this paper I will concentrate on a simplified presentation of connections between sciences dealing with cognition and humanistic aspect of technologies. Here we can separate many platforms which are the base for functioning of man who is trapped informative technologies. Cultural and social aspects are very important here. External informative world (and not only) overlaps internal world of human psychology. As we know, man has several psychological layers which are responsible for feelings, cultural archetypes and awareness. Adequate multimedial preparation and presentation of information can have a sublime impact on specific layers in human psychology, e.g. feelings, thus broadening this layer while narrowing the awareness. These techniques are used in advertising and public relations in order to make the customer buy a product and created a good image of a company. They are the activities which influence human psychology. Through experience and internal stimulus man is induced to certain behavior, including cultural behavior at a specific time and place. It is difficult to say whether it is good or bad but such is the situation and surely there are limits for such activities. Man himself can become resistant to these activities with time.

## 5 Conclusions

In spite of all significant and still increasing possibilities of multimedia we can assume that at the present stage of development multimedia and computers are only tools in human hands and, from this point of view, they can be used in different ways–according to their possibilities, destination and human decisions. In management practice they are mainly used at the operational level for formalized, structured, repeatable and programmable activities.

On the other hand if problems cannot be easily described by an algorithm or the phenomenon is relatively new the use of machines in decision-making is practically impossible at this stage of development. Here, manager's decisions are still needed; he can only support himself with modern technology (information transfer, data collecting and processing, etc.). It refers mainly to the strategic level of management where decisions are usually taken in risky, unsure and even turbulent conditions.

Mass spread of multimedia thanks to technological achievements, especially during the last 150 years have created real possibilities of their application to many aspects of human life, including management on a large scale, creating event the base for global management and global world economy where everything can be produced and sold. Also, exchange and spreading of technology and information can be done on a large scale at the enormous speed. It gives a chance but also, as it is in the case of important events, carries dangers. Everything depends on the use of these technologies and their perfection.

As the data in Table 1 shows, the saturation of multimedia is usually larger in highly-developed with a high GDP level per person. In these countries the number of mobile phones can be compared to the number of population. We can draw a conclusion that these means can serve a better development and functioning of the states, business organizations or people themselves, but it is not always so.

Surely, the use of e-business, and internet economy gives many possibilities and chances in economy or management but it also requires modern technology infrastructure which needs large financial investments. As it can be seen from the presented data, internet economy and e-business with the use of multimedia is best developed in the richest areas, e.g. the USA or EU where about 2/3 of the world's GDP is produced. However, non-material values, as in a typical information society, increase in value.

## References

1. Główczyk J.: *Uniwersalny słownik ekonomiczny*. Fundacja Innowacja, Warszawa (2000) 385–394
2. Steinman H, Schreyogg G.: *Zarządzanie, podstawy kierowania przedsiębiorstwem.* Politechnika Wrocławska, Wrocław (1992) 18–20.
3. Peszko A.: *Podstawy zarządzania organizacjami.* Wydawnictwo AGH, Kraków (1997) 51.
4. Stoner J., Wankel C.: *Kierowanie,* PWE, Warszawa (1992).
5. Siemieniecki B.: *Komputery i hipermedia w procesie edukacji dorosłyc.,* Wyd. A. Marszałek, Toruń (1998) 3–10.
6. Central Intelligence Agency (CIA) USA: `www.cia.gov` (The World Factbook), July 2005.