LATVIJAS UNIVERSITĀTES RAKSTI
ACTA UNIVERSITATIS LATVIENSIS

Datorzinātne un informācijas tehnoloģijas
Computer Science and Information Technologies

Databases and Information Systems

Doctoral Consortium

Edited by Rusins Freivalds

Sixth International Baltic Conference
BalticDB&IS 2004
Riga, Latvia, June 6-9, 2004

673

ISSN 1407-2157

LATVIJAS UNIVERSITĀTES
RAKSTI
673. SĒJUMS

Datorzinātne un informācijas tehnoloģijas

SCIENTIFIC PAPERS
UNIVERSITY OF LATVIA
VOLUME 673

Computer Science and Information Technologies

SCIENTIFIC PAPERS
UNIVERSITY OF LATVIA
VOLUME 673

Computer Science and Information Technologies

Databases and Information Systems

Doctoral Consortium

Edited by Rūsiņš Freivalds

Sixth International Baltic Conference
BalticDB&IS 2004
Riga, Latvia, June 6-9, 2004

LATVIJAS UNIVERSITĀTE

LATVIJAS UNIVERSITĀTES
RAKSTI
673. SĒJUMS

Datorzinātne un informācijas tehnoloģijas

Datu bāzes un informācijas sitēmas

Doktorantu konsorcijs
Red. Rūsiņš Freivalds

Sestā Starptautiskā Baltijas konfernce
BalticDB&IS 2004
Rīga, Latvija, 2004. gada 6.–9. jūnijs

LATVIJAS UNIVERSITĀTE

UDK 004
Da 814

Visi krājumā ievietotie raksti ir recenzēti

Pārpublicēšanas gadījumā nepieciešama Latvijas Universitātes atļauja
Citējot atsauce uz izdevumu obligāta

ISSN 1407-2157
ISBN 9984-770-12-5

Table of Contents

Conference Committee

Advisory Committee

Janis Bubenko, Sweden
Arne Solvberg, Norway

Programme Co-Chairs

Janis Barzdins, Latvia
Albertas Caplinskas, Lithuania
Rusins Freivalds, Latvia

Organising Co-Chairs

Juris Borzovs, Latvia
Inara Opmane, Latvia

Co-ordinators

Ahto Kalja, Estonia
Saulius Maskeliunas, Lithuania

Programme Committee

Witold Abramowicz, Poland
Mikhail Auguston, USA
Janis Bicevskis, Latvia
Johann Eder, Austria
Hans-Dieter Ehrich, Germany
Jorgen Fischer Nilsson, Denmark
Janis Grundspenkis, Latvia
Remigijus Gustas, Sweden
Hele-Mai Haav, Estonia
Igor Kabashkin, Latvia
Leonid Kalinichenko, Russia
Ahto Kalja, Estonia
Audris Kalnins, Latvia
Marite Kirikova, Latvia
Patrick Lambrix, Sweden
Jozef M. Zurada, USA
Saulius Maskeliunas, Lithuania
Mihhail Matskin, Norway
Boris Novikov, Russia
Monika Oit, Estonia
Algirdas Pakstas, UK
Jaan Penjam, Estonia
Jaroslav Pokorny, Czech Republic
Gunter Saake, Germany
Klaus-Dieter Schewe, New Zealand
Jaak Tepandi, Estonia
Bernhard Thalheim, Germany
Enn Tyugu, Estonia
Olegas Vasilecas, Lithuania
Benkt Wangler, Sweden
Mudasser Wyne, USA
Arkady Zaslavsky, Australia

Additional Referees

Peter Ahlbrecht, Germany
Arne Ansper, Estonia
Per Backlund, Sweden
Dmitry Barashev, Russia
Asa Dahlstedt, Sweden
Rainer Eckstein, Germany
Silke Eckstein, Germany
Janis Eiduks, Latvia
Mohamed Medhat Gaber, Australia
Mika Hirvensalo, Finland
Hagen Hoepfner, Germany
Vaida Jakoniene, Sweden
Paulis Kikusts, Latvia
Zealand
Markus Kirchberg, New Zealand
Christian Koncilia, Austria
Vahur Kotkas, Estonia
Algirdas Laukaitis, Lithuania
Marek Lehmann, Austria
Karl Neumann, Germany
Karlis Podnieks, Latvia
Oleg Proskurnin, Russia
Fike Schallehn, Germany
Ingo Schmitt, Germany
Asko Seeba, Estonia
Andrey Simanovsky, Russia
Eva Soderstrom, Sweden
Mattias Strand, Sweden
Alexei Tretiakov, New

Viljar Tulit, Estonia
Juris Viksna, Latvia

Local Organising Committee

Janis Barzdins
Janis Benefelds
Ansis Ataols Berzins
Janis Bicevskis
Uldis Bojars
Juris Borzovs, co-chair
Edgars Celms
Andrejs Dubrovskis
Anita Ermusa
Rusins Freivalds
Martins Gills
Ija Haritonova
Maksims Kravcevs
Davis Kulis
Lelde Lace
Ilvars Mizniks
Laila Niedrite
Inara Opmane, co-chair
Martins Opmanis
Raimonds Praude
Oksana Scegulnaja-Dubrovska
Darja Smite
Agnis Stibe
Maris Treimanis
Valdis Vitolins
Aleksandrs Zelenkovs
Janis Zuters

Conference Secretary

Edgars Celms, Latvia

Preface

The Baltic Conference on Databases and Information Systems is a biannual international forum for technical discussion among researchers and developers of database and information systems. The objective of the conference is to bring together researchers as well as practitioners and PhD students in the field of computing research that will improve the construction of future information systems. On the other hand, the conference is giving opportunities to developers, users and researchers of advanced IS technologies to present their work and to exchange their ideas and at the same time providing a feedback to database community.

The 6th International Baltic Conference on Databases and Information Systems (Riga, Latvia, June 6-9, 2004) is continuing the series of conferences that have been held in Trakai, Lithuania (1994), Tallinn, Estonia (1996, 2002), Riga, Latvia (1998), and Vilnius, Lithuania (2000).

All the accepted papers were selected by the Program Committee on the basis of referee reports. Each paper was reviewed by at least three referees who judged the papers for originality, quality, and consistency with the topics of the conference.

To activate and motivate students the organizers included Doctoral Consortium in the program of the conference. This is the first Doctoral Consortium in the Baltic Conferences on Databases and Information Systems. In spite of our fears the response from the international community of students was quite good. Program Committee was able to accept 25 by students (sometimes co-authored by their thesis advisers) from Estonia, Latvia, Lithuania, Norway and Russia. This allowed us to divide the student talks into two sections. Accordingly, we divided the papers in this volume of proceedings.

We thank all the contributors (successful and not so successful), the numerous referees, the members of Program Committee, and, of course, our sponsors, namely, IEEE Communications Society, Latvian Information Technology and Telecommunications Association (LITTA), DATI GRUPA, EXIGEN, UNIBANKA, it ALISE, LATTELEKOM, University of Latvia and VLDB Baltic Fund.

Rūsiņš Freivalds
Editor

QUANTUM COMPUTING AND COMPLEXITY

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004. VOLUME 673, Databases and Information Systems
13–17 P.

A Way Of Constructing Functions With A Low Polynomial Degree

Aleksandrs Belovs[1]

Department of Computer Science, University of Latvia,
29 Raina boulevard, Riga, Latvia
sd20006@lanet.lv

Abstract. In this paper we introduce a way of computing Boolean function polynomial's coefficients, using a special presentation of them. It helps in constructing functions with a low polynomial degree.
Keywords. Boolean functions, polynomial degree, decision trees.

1 Introduction

1.1 Zhegalkin polynomial

Let $F(x_1,\ldots,x_n)$ be a Boolean function. There exists a unique Zhegalkin polynomial (or function polynomial) $P_F(x_1,\ldots,x_n)$ such that

$$F(x_1,x_2,\ldots,x_n) = P_F(x_1,x_2,\ldots,x_n) \tag{1}$$

for all possible x_i values.

Boolean function polynomial has several applications in the theoretical computer science. We will give an example of its use in the theory of quantum decision trees.

A decision tree is an algorithm for computing $F(x_1,\ldots,x_n)$ that accesses $x_1,\ldots,x_n$ by asking questions about the values of x_i. The complexity of a query algorithm is the maximum number of questions that it asks. The query algorithm complexity of a function F is the minimum complexity of a query algorithm correctly computing F.

The theory of computation studies various models of computation: deterministic, non-deterministic, probabilistic and quantum (see [5] on traditional models of computation and [3] on quantum computation). Similarly, there are query algorithms of all those types [2].

We will use notation $D(F)$ for deterministic decision tree complexity of Boolean function F, and $Q_E(F)$ for exact quantum decision tree complexity (See [2] for definitions). Function exact degree $\deg(F)$ is the degree of polynomial P_F defined in (1).

It has been proved, that $D(F) \geq \deg(F)$ and $Q_E(F) \geq \frac{\deg(F)}{2}$. (The complexity of other types of query algorithms is similarly related to other notions of polynomial degree which are not considered in this paper.) Considering both of this inequalities one can see that in constructing Boolean functions that have an advantage in quantum computing, comparing to the deterministic one, functions such that $\deg(F) < D(F)$ seem interesting. But there are few functions satisfying this inequality.

[1] Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP).

1.2 Sensitivity

Let F be a Boolean function. Sensitivity of function F on input $(x_1, x_2, \ldots, x_n)$ is a number of $1 \leq i \leq n$ such that

$$F(x_1, \ldots, x_i, \ldots, x_n) \neq F(x_1, \ldots, 1 - x_i, \ldots, x_n).$$

Sensitivity of function $s(F)$ is defined as a maximum of sensitivities over all possible inputs. It has been proved in [2], that $D(F) \geq s(F)$. This fact we will use in this article.

2 Main Result

2.1 Function presentation

There are different ways of presenting Boolean functions. One we will mostly use in this article is following. We will fix an n-element set $U = \{1, 2, \ldots, n\}$, and present a set of functions $T \subseteq \{\{0\}, \{1\}, \{0, 1\}\}^U$. For each function $t \in T$ we will define a set of sequences $\pi(t) = \prod_{i \in U} t(i)$. Also we will use notation $\pi(T) = \bigcup_{t \in T} \pi(t)$.

Then function F will be defined as $\chi(\pi(T))$, that is:

$$F(x_1, \ldots, x_n) = \begin{cases} 1, \text{if } (x_1, \ldots, x_n) \in \pi(T) \\ 0, \text{in all other cases} \end{cases}. \tag{2}$$

We will write this presentation in a form of a table, where rows correspond to the functions, columns to the set U elements, and their intersections to the function values, using '-' for $\{0, 1\}$.

Example An example of a function:

$$F_T: \begin{array}{ccccccc} 0 & 0 & 0 & 1 & - & - & - \\ - & 0 & 0 & 0 & 1 & - & - \\ - & - & 0 & 0 & 0 & 1 & - \\ - & - & - & 0 & 0 & 0 & 1 \\ 1 & - & - & - & 0 & 0 & 0 \\ 0 & 1 & - & - & - & 0 & 0 \\ 0 & 0 & 1 & - & - & - & 0 \end{array} \tag{3}$$

This presentation is rather useful, because, in a case, when a number of functions is rather small, firstly, it is easy to compute a value of the function, secondly, as we will show, it is easy to calculate one specified polynomial's coefficient, without calculating the whole polynomial, if the function set complies with a condition:

Condition of no intersection

$$\forall t_1, t_2 \in T : (t_1 \neq t_2) \Rightarrow (\exists i \in U : t_1(i) \cap t_2(i) = \emptyset). \tag{4}$$

In the table this means that for each two rows there exists a column, such that in its intersections with both rows there stand '1' and '0' (in any order).

Lemma If a set of functions T satisfies condition of no intersection, than $\forall t_1, t_2 \in T : (t_1 \neq t_2) \Rightarrow (\pi(t_1) \cap \pi(t_2) = \emptyset)$.

Definition We will say, that function F is presented using a set of functions T, if there exist such U, that T and F satisfy (2) and moreover is fulfilled condition of no intersection (4).

Definition If $t \in T$, than we will use notation: $Z(t) = t^{-1}(\{0\})$ and $O(t) = t^{-1}(\{1\})$.

2.2 Theorem 1

If Boolean function F is presented using a set of functions T, than coefficient in P_F (polynomial. defined in (1)) at the $\prod_{i \in S} x_i$, where $S \subseteq U$, is equal to $\sum_{t \in T} \nu(t, S)$, where

$$\nu(t, S) = \begin{cases} (-1)^{|S \setminus O(t)|} & , \text{ if } O(t) \subseteq S \subseteq O(t) \cup Z(t) \\ 0 & , \text{ in all other cases} \end{cases}$$

Proof If $S \subseteq U$, than by $f(S)$ we will understand $F(x_1, x_2, \ldots, x_n)$, where $x_i = 1$ if and only if $i \in S$. Notation $p(S)$ we will use for a coefficient in P_F at the $\prod_{i \in S} x_i$. Obviously, $f(S) = \sum_{R \subseteq S} p(R)$. Using the method of inclusion and exclusion (See, for example, [6]), (2) and Lemma, we have

$$p(S) = \sum_{R \subseteq S} (-1)^{|S \setminus R|} f(R) = \sum_{R \subseteq S \cap \pi(T)} (-1)^{|S \setminus R|} = \sum_{t \in T} \sum_{R \subseteq S \cap \pi(t)} (-1)^{|S \setminus R|}.$$

If $O(t) \not\subseteq S$, we have $S \cap \pi(t) = \emptyset$. Else

$$\sum_{R \subseteq S \cap \pi(t)} (-1)^{|S \setminus R|} = (-1)^{|S \setminus O(t)|} \delta_0^{|S \cap t^{-1}(\{0,1\})|},$$

where δ stands for the Dirac function. Here we use a well-known formula from combinatorics: $\sum_{A \subseteq B} (-1)^{|A|} = \delta_0^{|B|}$.

Summing it all up, we have the required.

As a usage of this theorem we will prove

2.3 Theorem 2

If Boolean function F is presented using a set of functions T, than

$$\deg(F) \leq \max_{t \in T} |O(t) \cup Z(t)|.$$

3 Some Examples And Future Work

So, one can see that for a specified in (3) function F_7, $\deg(F_7) \leq 4$. In fact it is equal to 4, what can be checked, calculating a coefficient at the $x_1x_2x_3x_4$. But on the other side, when all x_i is equal to 0, sensitivity is 7, so $D(F_7) = 7$. In a same way it is possible to construct a function F_{2n+1} with $\deg(F_{2n+1}) = n + 1$ and $D(F_{2n+1}) = 2n + 1$.

This example is the best one that can be achieved, using only techniques of sensitivity and Theorem 2. So, constructing functions with a larger gap between $D(F)$ and $\deg(F)$, it is necessary to use Theorem 1 more carefully or to use other estimates of $D(F)$, because a question of easy $D(F)$ computing for a wide class of functions is still open.

Let us illustrate the said with two examples.

3.1 Example 1

0	0	0	0	1	–	–	–	–
0	0	0	–	0	1	–	–	–
0	0	0	1	–	0	–	–	–
1	1	1	0	1	–	–	–	–
1	1	1	–	0	1	–	–	–
1	1	1	1	–	0	–	–	–
–	–	–	0	0	0	0	1	–
–	–	–	0	0	0	–	0	1
–	–	–	0	0	0	1	–	0
–	–	–	1	1	1	0	1	–
–	–	–	1	1	1	–	0	1
–	–	–	1	1	1	1	–	0
0	1	–	–	–	–	0	0	0
–	0	1	–	–	–	0	0	0
1	–	0	–	–	–	0	0	0
0	1	–	–	–	–	1	1	1
–	0	1	–	–	–	1	1	1
1	–	0	–	–	–	1	1	1

This function has sensitivity 9 (when all $x_i = 0$). So $D(F) = 9$. There is no row, such that summary number of '0' and '1' in it is larger than 5. But it is easy to check, that for every set of 5 columns there is 2 possibilities:

- There is no row, such that on its intersections with selected columns there is no '–'.
- There are exactly 2 such rows, and a number of '1' in one of them is 1, and in another one 4.

Using theorem 1, we can see, that $\deg(F) \leq 4$. In fact $\deg(F) = 4$.

Note, that the rows with only one '1' in them present a function with sensitivity 9 and polynomial degree 5.

3.2 Example 2

An improved version of F_7.

0	0	0	1	–	–	–	0
–	0	0	0	1	–	–	0
–	–	0	0	0	1	–	0
–	–	–	0	0	0	1	0
1	–	–	–	0	0	0	0
0	1	–	–	–	0	0	0
0	0	1	–	–	–	0	0
1	1	1	0	–	–	–	1
–	1	1	1	0	–	–	1
–	–	1	1	1	0	–	1
–	–	–	1	1	1	0	1
0	–	–	–	1	1	1	1
1	0	–	–	–	1	1	1
1	1	0	–	–	–	1	1

In a same way, it has been done in example 1, one can check that $\deg(F) = 4$.

If a deterministic decision tree asks a value of one of $x_1, \ldots, x_7$, we answer '0', while there are still some unknown x_i from this set. If the decision tree asks x_8, we answer '0', and reduce to the case of F_7. If it asks a value of the last x_i from $x_1, \ldots, x_7$, we answer '1'. So $D(F) = 8$.

3.3 Resume

The technique, introduced in this article, makes it possible the use of standard combinatorics in constructing functions with a low polynomial degree.

It helps to construct a large class of such functions, among them there can be one with a small $Q_E(F)$. It is possible, that with some additional conditions, one will be able to guarantee small $Q_E(F)$.

It is interesting to construct functions with a larger gap between $\deg(F)$ and $D(F)$, than those, that are known. Generalization of theorem 2 or another way of presenting Boolean functions can be helpful in this question.

References

[1] Ambainis, A., Freivalds, R. (2003) Boolean function with a low polynomial degree. Latvijas Zinatnu Akademijas Vestis, B dala, vol. 57, No. 3/4 (626/627), pp. 74-77.

[2] Burhman, H., de Wolf, R. (2001) Complexity measures and decision tree complexity: a survey. Theoretical Computer Science, 288, pp. 21-43.

[3] Gruska, J. (1999) Quantum computing. McGraw Hill. London, 439 pp.

[4] Freivalds, R., Miyakawa, M., Rosenberg, I. (2003) Complexity of decision trees for Boolean functions. ISMVL 2003: pp. 253-257.

[5] Papadimitriou, C. (1994) Computational complexity. Addison-Wesley. Reading. 500 pp.

[6] Stanley, R. (1986) Enumerative combinatorics. Volume I, Wadsworth, California.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004. VOLUME 673, Databases and Information Systems
18–24 P.

Quantum lower bounds for the set equality problems

Gatis Midrijānis

University of Latvia,
29 Raina boulevard, Riga, Latvia
gatis@zzdats.lv

Abstract

The set equality problem is to decide whether two sets A and B are equal or disjoint, under the promise that one of these is the case. Some other problems, like the Graph Isomorphism problem, is solvable by reduction to the set equality problem. It was an open problem to find any $w(1)$ query lower bound when sets A and B are given by quantum oracles with functions a and b.

We will prove $\Omega(\frac{n^{1/3}}{\log^{1/3} n})$ lower bound for the set equality problem when the set of the preimages are very small for every element in A and B.

1 Introduction, Motivation and Results

The Shor's integer factoring quantum algorithm provides exponential speed-up over the best known classical algorithm. This motivates to search other quantum algorithms with great speed-up. However, proving quantum lower bounds for such problems is not trivial, for example, proving the exponential quantum lower bound for NP-Complete problems will imply $P \neq NP$.

One of the problems quantum computer could have an exponential speed-up over classical computer is the Graph Isomorphism problem. One way to attack this problem could be by the reduction to the set equality problem. Notice the sets of all permutations over vertexes for given graphs. If these sets are equal, then there is an isomorphism between the graphs, but if there is not isomorphism between graphs, then these sets are strictly disjoint.

Denote the set $\{1, 2, ..., n\}$ by $[n]$.

Definition 1 *Let $a : [n] \mapsto [m]$ and $b : [n] \mapsto [m]$ be a functions. Let A be a set of all $a's$ images $A = \{a(1), a(2), ..., a(n)\}$ and $B = \{b(1), b(2), ..., b(n)\}$. There is a promise that either $A = B$ or $A \cap B = \emptyset$.*

Call ***the general set equality*** *problem to distinguish these two cases.*

Finding quantum query lower bound for general set equality problem was posed an open problem by Shi[12].

We will show that Ambainis' [2] adversary method imply $\Omega(\sqrt{n})$ lower bound for the general set equality problem. The proof uses the possibility to have many preimages for some image. However, graph theorists think that the Graph Isomorphism problem, when graphs are promised not to be equal with themselves by any nonidentical permutation, still is very complex task. Now reduction lead us to the set equality where a and b are one-to-one functions.

Definition 2 *Call the general set equality problem to be a* ***one-to-one set equality*** *problem if* $a(i) \neq a(j)$ *and* $b(i) \neq b(j)$ *for all* $i \neq j$.

The proof that worked for the general set equality problem does not work for one-to-one set equality problem, because it uses that fact that there can be very many preimages for any element of the sets. However, we will prove lower bound for a problem between these problems.

Definition 3 *Call the general set equality problem to be a* ***f(n) set equality*** *problem if* $|a^{-1}(x)| = O(f(n))$ *and* $|b^{-1}(x)| = O(f(n))$ *for all images* $x \in a([n]) \cup b([n])$ *and for some function* f.

We will prove $\Omega(\frac{n^{1/3}}{\log^{1/3} n})$ lower bound for the $log(n)$ set equality problem.

The first result for lower bounds of the set equality like problem was done by Aaronson [1]. He showed $\Omega(n^{1/6})$ lower bound for so called set comparison problem: to decide whether two sets are equal or disjoint on a constant fraction of elements. He also assumed that both a and b are one-to-one functions. In this paper, we will study lower bound of problem when these sets A and B are strictly disjoint or equal, however a and b is not a one-to-one.

2 Preliminaries

2.1 Quantum Query algorithms

The most popular model of quantum computing is a query (oracle) model where the input is given by a black box. For more details, see a survey by Ambainis [3] or textbook by Gruska [8]. In this paper we are able to skip them because our proof will be built on reduction to solved problems.

One of the most amazing quantum algorithms is a Grover's search algorithm. It shows how a given $x_1 \in \{0,1\}, x_2 \in \{0,1\}, \ldots, x_n \in \{0,1\}$ to find the i that $x_i = 1$ with $O(\sqrt{n})$ queries.

This algorithm can be generalized to so called amplitude amplification [7]. Using amplitude amplification one can make good quantum algorithms for many problems till the quadratic speed-up over classical algorithms.

By straightforward use of amplitude amplification we get a quantum algorithm with $O(\sqrt{n})$ queries for the general set equality problem and a quantum algorithm with $O(n^{1/3})$ queries for the one-to-one set equality problem.

2.2 Quantum query lower bounds

There are two main approaches to get good quantum lower bounds. The first is Ambainis' [2] quantum adversary method. The other is lower bound by polynomials introduced by Beals et al. [5] and substantially generalized by Aaronson [1] and Shi [12]. Although explicitly we will use only Ambainis' method, main result we will get by a reduction to problem, solved by polynomials' method.

The basic idea of adversary method is that, if we can construct relation $R \subseteq A \times B$, where A and B consisting of 0-instances and 1-instances and there is a lot of ways how to get from an instance in A to an instance in B that is in the relation and back by flipping various variables, then query complexity must be high.

Theorem 1 *[2] Let $f(x_1, \ldots, x_N)$, be a function of n $\{0, 1\}$-valued variables and X, Y be two sets of inputs such that $f(x) \neq f(y)$ if $x \in X$ and $y \in Y$. Let $R \subset X \times Y$ be such that*

- *For every $x \in X$, there exist at least m different $y \in Y$ such that $(x, y) \in R$.*
- *For every $y \in Y$, there exist at least m' different $x \in X$ such that $(x, y) \in R$.*
- *For every $x \in X$ and $i \in \{1, \ldots, n\}$, there are at most l different $y \in Y$ such that $(x, y) \in R$ and $x_i \neq y_i$.*
- *For every $y \in Y$ and $i \in \{1, \ldots, n\}$, there are at most l' different $x \in X$ such that $(x, y) \in R$ and $x_i \neq y_i$.*

Then, any quantum algorithm computing f uses $\Omega(\sqrt{\frac{mm'}{ll'}})$ queries.

2.3 The collision problem

Finding $w(1)$ quantum lower bound for the collision problem was an open problem since 1997. In 2001 Scott Aaronson [1] solved it showing polynomial lower bound. Later his result was improved by Yaoyun Shi [12]. Newly Shi's result was extended by Samuel Kutin [10] and by Andris Ambainis [4] in another directions.

Below is exact formulation of collision problem due to Shi [12].

Definition 4 *Let $n > 0$ and $r \geq 2$ be integers with $r|n$, and let a function of domain size n be given as an oracle with the promise that it is either one-to-one or r-to-one. Call the* ***r-to-one*** *collision problem the problem to distinguishing these two cases.*

Shi [12] showed quantum lower bound for r-to-one collision problem.

Theorem 2 *[12] Any error-bounded quantum algorithm to solve r-to-one collision must evaluate the function $\Omega((n/r)^{1/3})$ times.*

3 Results

3.1 Lower bound for the general set equality problem

Theorem 3 *Any quantum algorithm which solves the general set equality problem makes* $\Omega(\sqrt{n})$ *queries.*

Proof. Simple use of Ambainis' Theorem 1. Since Ambainis' Theorem 1 deals with boolean functions, we will modify any quantum algorithm that solves the general set equality problem to an algorithm, that computes boolean function.

We will prove this theorem even in a restricted case, when functions returns only two values, let say 0 and 1. So we have a problem, given two functions $a : n \mapsto \{0,1\}$ and $b : n \mapsto \{0,1\}$ answer either the sets $A = \{a(1), ..., a(n)\}$ and $B = \{b(1), ..., b(n)\}$ are equal or disjoin under the promise that one of this is the case.

Let $f : 2n \mapsto \{0,1\}$ be a partially defined function. ,

$$f(a(1), a(2), ..., a(n), b(1), b(2), ..., b(n)) = \begin{cases} 1, & \text{if } A = B; \\ 0, & \text{if } A \cap B = \emptyset. \end{cases}$$

It is easy to see, that if we can solve a general set equality problem, we can compute this function with constant slowdown, too.

Let construct the relation R from Ambainis' Theorem 1 as follows:

$$R = \{(0^n 1^n,\ 0^i 10^{n-i-1} 1^i 01^{n-i-1}) : 0 \leq i < n\}.$$

One can check that R is well defined and $m = n$, $m' = 1$, $l = 1$ and $l' = 1$. Thus any quantum algorithm computing f uses $\Omega(\sqrt{n})$ queries. □

3.2 Lower bound for the log(n) set equality problem

Now we will prove the main result in this paper.

Theorem 4 *Any error-bounded quantum algorithm which solves the log(n) set equality problem makes* $\Omega(\frac{n^{1/2}}{\log^{1/3} n})$ *queries.*

Proof:

To prove Theorem 4 we will reduce r-to-one collision problem to the log(n) set equality problem. We are given function $f : [n] \mapsto [m]$, under promise to be either r-to-one or one-to-one and $r = \lceil \log n \rceil$ and $r|n$. We randomly choose two sets A and B such that $|A| = |B| = n/2$ and $A \cup B = [n]$ and $A \cap B = \emptyset$. Denote $A' = f(A)$ and $B' = f(B)$. It is obviously that if f is one-to-one then $A' \cap B' = \emptyset$.

If f is r-to-one then the situation is more complicate. In the next subsection we will prove that with big probability holds that A' and $B's$ includes all images of f, thus $A' = B' = f([n])$.

Let the functions a and b from Theorem 4 be the same as f but domain for a is A and domain for b is B.

Denote the set of all preimages of x in the set A by $f^{-1}{}_A(x) = f^{-1}(x) \cap A$. Since $|f^{-1}(x)| = r$ for every $x \in f([n])$, it is clear that also $|a^{-1}(x)| = O(\log n)$ and $|b^{-1}(x)| = O(\log n)$ for every $x \in f([n])$.

So with constant probability we get the log(n) set equality problem with domain size $n/2$. Now Theorem 2 implies Theorem 4. □

3.3 Reduction

Lemma 1 *From all possible divisions of a set $[n]$ into two equal sized parts A and B such that $A \cap B = \emptyset$, only few of them are such that for some $x \in f([n])$ there is no preimage either in A or B.*

Proof:

Total count of all (possibly uniform) divisions are

$$C_n^{n/2} = \frac{n!}{(n/2)!(n/2)!}.$$

Total count of such divisions is at most the count of images (n/r) multiplied by count of divisions where one fixed $x \in f([n])$ has no preimage either in A or B.

Assume that all preimages of x is in A, thus B has not any of them. Number of ways how we can choose residual elements is

$$C_{n-r}^{n/2-r} = \frac{(n-r)!}{(n/2-r)!(n/2)!}.$$

Analysis of an opposite assumption is similar, so probability to choose division which is bad on x is

$$\frac{2C_{n-r}^{n/2-r}}{C_n^{n/2}} = \frac{2(n-r)!(n/2)!(n/2)!}{(n/2-r)!(n/2)!n!} = \frac{2(n/2)(n/2-1)(n/2-2)...(n/2-r+1)}{n(n-1)(n-2)...(n-r+1)} =$$

$$= \frac{n/2-1}{n-1}\frac{n/2-2}{n-2}...\frac{n/2-r+1}{n-r+1} \leq (\frac{1}{2})^{r-1}.$$

So probability to choose bad division for any $x \in f([n])$ is at most

$$(\frac{1}{2})^{r-1}\frac{n}{r} = \frac{2n}{2^r r}.$$

Since $r = \lceil \log n \rceil$

$$\frac{2n}{2^r r} \leq \frac{2}{\lceil \log n \rceil}$$

which is small for large n.

□

4 Conclusion

Finding lower bound for the set equality problem is one of the most challenging today's task in theory of quantum query lower bounds. We have solved this problem partially. One can argue that to solve the set equality problem can be easier when functions are promised to be with a small range of preimages for all images. Our paper shows that the difference between the general set equality problem and the $\log n$ set equality problem is very small, respectively $\Omega(\sqrt{n})$ and $\Omega(\frac{n^{1/3}}{\log^{1/3} n})$. This enforce opinion that quantum computer probably cannot solve the one-to-one set equality problem with only polylogarithmic number of questions.

5 Acknowledgments

I am very grateful to Andris Ambainis for introducing me with quantum algorithms and also with this problem as well as comments during writing this paper.

Research supported by Grant No.01.0354 from the Latvian Council of Science, and Contract IST-1999-11234 (QAIP) from the European Commission.

References

[1] S. Aaronson. Quantum lower bound for the collision problem. *In Proceedings Proceedings of ACM STOC'2002, pp. 635-642, 2002.* quant-ph/0111102.

[2] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64:750-767, 2002. Earlier versions at STOC'00 and quant-ph/0002066.

[3] A. Ambainis. Quantumm query algorithms and lower bounds. *Proceedings of FOTFS III. to appear.*

[4] A. Ambainis. Quantum lower bounds for collision and element distinctness with small range, 2003. quant-ph/0305179

[5] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf. Quantum lower bounds by polynomials. *Journal of ACM*, 48: 778-797, 2001. Earlier version at FOCS'98.

[6] A. Berzina, A. Dubrovsky, R. Freivalds, L. Lace, O. Scegulnaja. Quantum query complexity for some graph problems. SOFSEM 2004:140-150.

[7] G. Brassard, P. Høyer, M. Mosca and A. Tapp. Quantum amplitude amplification and estimation. *to appear in AMS Contemporary Mathematics Series Millennium Volume entitled "Quantum Computation & Information".*

[8] J. Gruska. Quantum computing. McGraw-Hill. 1999.

[9] P. Hoyer, J. Neerbek. Y. Shi. Quantum lower bounds of ordered searching, sorting and element distinctness. *Algorithmica*, 34:429-448. 2002. Earlier versions at ICALP'01 and quant-ph/0102078.

[10] S. Kutin. Quantum lower bound for the collision problem, 2003.

[11] Y. Shi. Lower bounds of quantum black-box complexity and degree of approximating polynomials by influence of Boolean variables. *Information Processing Letters* 75:79-83, 2000.

[12] Y. Shi. Quantum lower bounds for the collision and the element distinctness problems *Proceedings of the 43rd Annual Symposium on the Foundations of Computer Science*, pp. 513-519, 2002.

[13] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *In proceedings: 35th Annual Symposium on Foundations of Computer Science, November 20-22, 1994, Santa Fe, New Mexico*, pages 124-134, IEEE Computer Society Press, 1994.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004, VOLUME 673, Databases and Information Systems
25 30 P.

Boolean Functions Of Low Polynomial Degree

Pēteris Ledinš, Rihards Opmanis[1]

Department of Computer Science, University of Latvia,
29 Raina boulevard, Riga, Latvia
ledins@latnet.lv, rixix@navigators.lv

Abstract. Boolean functions of high deterministic query complexity and low degree of representing polynomial have different applications in theoretical computer science, yet not many are known with such properties. We analyze some previously known and construct several new functions with such properties.
Keywords. Boolean functions, polynomial degree, decision trees.

1. Introduction

Boolean functions being quite simple in definition as returning results from $\{a, b\}^n$ to $\{a, b\}$, where $a \neq b$ can be chosen, are a subject to careful investigation as they can be used in proving different characteristics of different computational models.

For example, Boolean functions can be used to prove different results considering quantum decision tree complexity. If we use $deg(f)$ as the degree of multilinear polynomial representing Boolean function f, and $D(f)$ as the deterministic decision tree complexity of f then we have $D(f) \geq deg(f)$ [1]. For exact quantum decision tree complexity there exists a result $Q_E(f) \geq \frac{deg(f)}{2}$ [6].

For use in quantum computing the challange is to find a function with high $D(f)$ and low $deg(f)$, that could give some advantages in proving bound-related problems.

2. Notation

As noted before we will use $D(f)$, but it is easier to determine the maximum sensitivity of a Boolean function - $s(f)$. Sensitivity of f on input $(x_1, x_2, \ldots, x_n)$ is the number of of variables x_i with property that $f(x_1, x_2, \ldots, x_i, \ldots, x_n) \neq f(x_1, x_2, \ldots, 1 - x_i, \ldots, x_n)$

It has been proved that $s(f) \leq D(f)$ [2]. So, if we know $s(f)$ then we know that $D(f)$ is at least the same.

To describe Boolean functions we use a description in form of tables with each row for different type of input and each column for each variable. Each cell contains "1", "0" or "−" in each cell. A table is constructed to show all possible inputs giving either 1 or 0 as the result of function. "0" or "1" in ith column symbolizes that input defined by the row must have $x_i = 0$ or $x_i = 1$ resp. while "−" means that the value of x_i is not significant for the input.

[1] Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP).

3. Existing results

Currently there are several functions providing $D(f) > deg(f)$ that are used to create (e.g. iterations and other various operations) others.

1. Function $f_1(x_1, x_2, x_3)$ being 0 iff all variables are equal [1]. $D(f_1) = 3$, $deg(f_1) = 2$
2. Function $f_2(x)$ that equals 1 iff $x = x_1x_2x_3x_4$ equals 0011, 0100, 0101, 0111, 1000, 1010, 1011, or 1100 with $deg(f_2) = 2$ and $D(f_2) = 3$ [3].
3. Function $f_3(x_1, x_2, x_3, x_4, x_5, x_6)$ with $deg(f_3) = 3$ and $D(f_3) = 6$ by Kushilevitz, cited by [4]. f_3 equals 0 when sum of x_i equals 0, 4 or 5 or 3 and one of the following is true: $x_1 = x_2 = x_3 = 1$, $x_2 = x_3 = x_4 = 1$, $x_3 = x_4 = x_5 = 1$, $x_4 = x_5 = x_1 = 1$, $x_5 = x_1 = x_2 = 1$, $x_1 = x_3 = x_6 = 1$, $x_1 = x_4 = x_6 = 1$, $x_2 = x_4 = x_6 = 1$, $x_2 = x_5 = x_6 = 1$, $x_3 = x_5 = x_6 = 1$. Otherwise the value is 1.
4. Function $f_4(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ with $D(f) = 7$ and $deg(f) = 4$ from [5]. The value of function is 1 iff input is defined in Table 1.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
–	1	–	0	–	0	0
–	–	0	–	0	0	1
–	0	1	0	0	–	–
0	–	0	0	–	1	–
1	0	0	–	–	–	0
0	0	–	–	1	0	–
0	–	–	1	0	–	0

Table 1: Function f_4 from [5]

4. Construction of Boolean functions

In our search for Boolean functions of specified qualities we use the previously mentioned way of description and analyze regularities. For example, some kind of regularities can be seen in Table 1.

4.1. Hadamard matrices

A Hadamard matrix is a square matrix containing only $1s$ and $-1s$ such that when any two columns or rows are placed side by side, half the adjacent cells are the same sign and half the other (excepting from the count an L-shaped "half-frame" bordering the matrix on two sides which is composed entirely of $1s$) [7]. It is easy to see that one can change the order of columns and rows, still keeping the property of Hadamard matrices.

We use Hadamard matrix of order 8 from [8] shown in Table 2.

1	1	1	1	1	1	1	1
−1	−1	−1	1	−1	1	1	1
−1	1	−1	−1	1	−1	1	1
−1	1	1	−1	−1	1	−1	1
−1	1	1	1	−1	−1	1	−1
−1	−1	1	1	1	1	−1	1
−1	1	−1	1	1	1	−1	−1
−1	−1	1	−1	1	1	1	−1

Table 2: Hadamard matrix had.8.1 from [8]

4.2. Analysis of f_4

Table 1 can be constructed easily from Table 2 - first column and first row are to be removed, matrix has to be read from the right side. -1s replaced with 0s and 1s with "- " signs (as in Table 3). After that we only have to insert 1s in appropriate positions.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
–	–	–	0	–	0	0
–	–	0	–	0	0	–
–	0	–	0	0	–	–
0	–	0	0	–	–	–
–	0	0	–	–	–	0
0	0	–	–	–	0	–
0	–	–	–	0	–	0

Table 3: Building function f_4

First we want the definition of function to satisfy the following property:

Property 1 (Rihard's property) *If there exists* 1 *in cell* (i, j) *and* 1 *in cell* (k, l) *then exactly one cell from* $\{(i, l), (k, j)\}$ *must hold* 0 *and exactly one* - "–".

This property is a way to:

1. Allow each input be defined with exactly one row
2. Have some kind of minimal distance between each two rows, thus having a small amount of 1s and 0s used in the definition.

As one can see the definition of f_4 satisfies Property 1. Of course, there are several ways to put the 1s in the matrix of a function. A very regular way is moving the first row of Table 3 to the bottom and putting 1s on the diagonal as in Table 4.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
–	–	0	–	0	0	1
–	0	–	0	0	1	–
0	–	0	0	1	–	–
–	0	0	1	–	–	0
0	0	1	–	–	0	–
0	1	–	–	0	–	0
1	–	–	0	–	0	0

Table 4: Function f_5

Lemma 1 $s(f_5) = 7$

Lemma 1 is provided by sensitivity on zero input.

Theorem 1 $D(f_5) = 7$

Theorem 3 straight from Lemma 1.

Theorem 2 $deg(f_5) = 4$

Theorem 2 is proved by construction.

The number of members of the representing polynomial for f_5 is the same as for f_4 and equals 56 that shows how similar both the functions are.

4.3. More Boolean functions

Using the method showed previously we construct two other Boolean functions with similar properties representing the inputs giving 1 as result in Table 5 and Table 6. It must be reminded that f_6 is the same function as f_1.

x_1	x_2	x_3
0	–	1
–	1	0
1	0	–

Table 5: Function f_6

Theorem 3 $D(f_7) = 11$, $deg(f_7) = 6$

Deterministic query complexity comes from sensitivity and degree is found by construction.

Using the same method we can construct Boolean functions for other numbers of variables - 19, 23, 31, 43, 47 - using Hadamard matrices available from [8]. However,

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
0	–	0	0	0	–	–	–	0	–	1
–	0	0	0	–	–	–	0	–	1	0
0	0	0	–	–	–	0	–	1	0	–
0	0	–	–	–	0	–	1	0	–	0
0	–	–	–	0	–	1	0	–	0	0
–	–	–	0	–	1	0	–	0	0	0
–	–	0	–	1	0	–	0	0	0	–
–	0	–	1	0	–	0	0	0	–	–
0	–	1	0	–	0	0	0	–	–	–
–	1	0	–	0	0	0	–	–	–	0
1	0	–	0	0	0	–	–	–	0	–

Table 6: Function f_7

we have not constructed polynomials to find the degree of theese matrices, but still we can do some estimates using the method described in [9]. That gives the degree of representing polynomial $deg(f) = \frac{n-1}{2}$ for these functions.

5. Future work

The new functions found do have characteristics demanded, but still for these functions $deg(f) > \frac{n}{2}$, when $D(f) = n$, where n - the number of variables. A result needed is some function f_x with $deg(f) < \frac{n}{2}$ that could possibly give some quantum query algorithm that had better advantages against classical counterpart than any now known.

Besides that Hadamard matrices and the likes could provide with interesting Boolean functions and thus need deeper investigation.

References

[1] N. Nisan and M. Szegedy, On the degree of Boolean functions as real polynomials Computational Complexity, 4(4), pp. 301-313, 1994. Earlier version in STOC'92.

[2] H. Buhrman and R. de Wolf. Complexity Measures and Decision Tree Complexity: A Survey. In Theoretical Computer Science, 288:21-43, 2002.

[3] A. Ambainis, Polynomial degree vs. quantum query complexity, Proceedings of FOCSŠ2003, pages 230-239

[4] N. Nisan and A. Wigderson, On rank vs. communication complexity, Proc. of FOCSŠ94, pp. 841-836, 1994.

[5] A. Ambainis, R. M. Freivalds, Boolean function with a low polynomial degree, Proceedings of the Latvian Academy of Sciences, vol. 57, 2003, pages 74-77.

[6] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf. Quantum lower bounds by polynomials. Journal of ACM, 48: 778-797, 2001.

[7] E. W. Weisstein, Hadamard Matrix From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/HadamardMatrix.html

[8] N. J. A. Sloane, A Library of Hadamard Matrices,
http://www.research.att.com/~njas/hadamard/

[9] A. Belovs, A Way Of Constructing Functions With A Low Polynomial Degree, 2004

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673. Databases and Information Systems
31-39 P.

Constructing Boolean functions with a low polynomial degree.

Raitis Ozols*

University of Latvia, Faculty of Physics and Mathematics
Raiņa bulvāris 29, Rīga, Latvija, LV - 1459
sm01022@lanet.lv

Abstract. Because of development in quantum computing and other issues of theoretical computer science, there is a necessity for Boolean functions with a low polynomial degree, but high deterministic complexity. For such functions it is highly probable that quantum algoritms can compute them more quickly than deterministic algorithms. Yet only a few such functions are known. It is also known that we can derive functions with a greater number of variables from any such functions by doing simple iterations. These derived functions also have a low polinomial degree and high deterministic complexity.
In this paper we will show a technique for constructing such functions for a small number of variables and more complex iterations for constructing functions with higher number of variables.

Keywords. Boolean functions, low polynomial degree.

1. Introduction.

Boolean function is function $f(x_1; x_2; \ldots; x_n)$: $\{0; 1\}^n \rightarrow \{0; 1\}$. Decision tree of this function is algorithm, which asks values of variables x_i, and according to the responses calculates the value of function f. Complexity of this decision tree is the maximal number of questions that the algorithm has to ask. Complexity of decision tree of function f or deterministic complexity is the smallest complexity of decision tree which correctly calculates function f. In this paper we will denote the deterministic complexity of function f by $D(f)$.

For every Boolean function $f = f(x_1; x_2; \ldots; x_n)$ there is a polynom $p(x_1; x_2; \ldots; x_n)$ such that for all $x_i \in \{0; 1\}$ $p(x_1; x_2; \ldots; x_n) = f(x_1; x_2; \ldots; x_n)$. This polynom which *represents* a Boolean function, is unique and its degree is also the degree of the function. We will denote the degree with $deg(f)$. It is known that for every Boolean function f $deg(f) \leq D(f)$. Also it is easy to understand that $D(f) \leq n$ is always true, where n is the number of variables of function f (because if we know values of all variables of function we can always to compute the value of function f).

In this paper the set of all Boolean functions which have n variables, deterministic complexity is m and degree of corresponding polynomial is k, will be denoted with $B\langle k; m; n\rangle$. Its easy to see that $k \leq m \leq n$. Of course, we consider

* Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP)

functions with non fictive variables. Further we focus on functions which have $m = n$, those are $B\langle k; n; n\rangle$ type functions.

Currently there are some such functions f known, have $D(f)>deg(f)$:

1. The function $f(x_1; x_2; x_3) \in B\langle 2; 3; 3\rangle$. Proposed by Nisan and Szegedy (1994).
2. The function $f(x_1; x_2; x_3; x_4) \in B\langle 2; 3; 4\rangle$. Proposed by Ambainis (2003). See [2].
3. The function $f(x_1; x_2; x_3; x_4; x_5; x_6) \in B\langle 3; 6; 6\rangle$. Proposed by Kushilevitz (unpublished result quoted in Nisan and Widgerson (1995)).
4. The function $f(x_1; x_2; x_3; x_4; x_5; x_6; x_7) \in B\langle 4; 7; 7\rangle$. Proposed by Ambainis and Freivalds (2003). See [1].

Remark. Construction of 3rd and 4th function related to block designs.

2. Constructing of new function $f \in B\langle 4; 7; 7\rangle$.

Now let us show how to construct function $f_0(x_1; x_2; x_3; x_4; x_5; x_6; x_7) \in B\langle 4; 7; 7\rangle$. Later we will see that this function is different from the function proposed by Ambainis and Freivalds in [1].

Construction. Let us write variables $x_1, x_2, \ldots, x_7$ where $x_i \in \{0; 1\}$ (Figure 1) in a circle and consider them as graph vertices. Let us compare variables x_1 and x_2. If $x_1 = x_2$ then we connect them with a continuous line —— . If $x_1 \neq x_2$ then we connect them with dashed line - - - . After that we compare x_2 and x_3 and again connect them with the appropriate line. We continue util we get variables x_7 and x_1 which we again connect. This is the way how we get closed cyclic graph with "coloured" edges (Figure 2). Edges which are drawn with dashed lines let us call "differences". Now we show that number of differences will always be an even number. Sum of $(x_1 - x_2) + (x_2 - x_3) - \ldots + (x_6 - x_7) + (x_7 - x_1) = 0$ is an even number, edges which have equal numbers in their endings correspond to difference 0, but edges which have different numbers in their endings correspond to diference

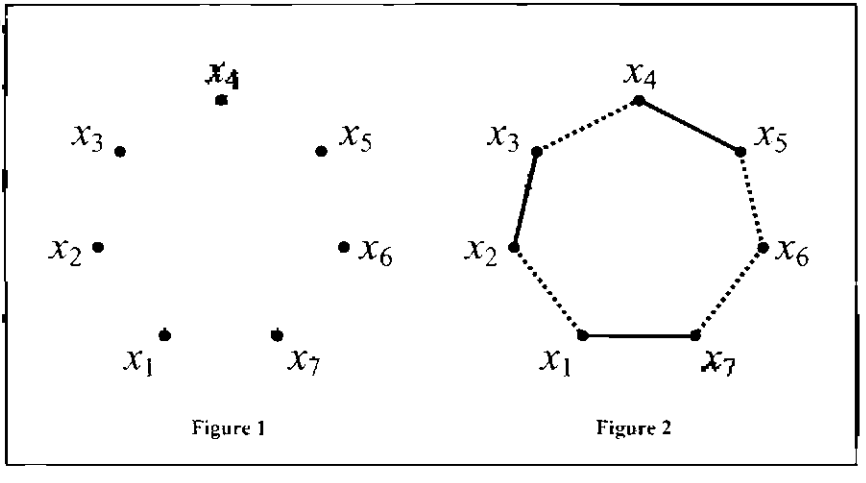


Figure 1 Figure 2

which is odd number. Therefore the sum of some odd numbers, number of which is equal to the number of differences, will be an even number. Therefore the number of these odd numbers and also the number of diferences can only be an even number. Even numbers which are not greater than 7 are 0, 2, 4 and 6, therefore the number of differences can only be 0, 2, 4 or 6. It can be verified that the number of differences can take any of those values, therefore none of them can be discarded.

Further it is easy to understand that the number of differences in the graph we can expressed by a function

$$\varphi = (x_1 - x_2)^2 + (x_2 - x_3)^2 + \ldots + (x_6 - x_7)^2 + (x_7 - x_1)^2.$$

Therefore for all $x_1, x_2, \ldots, x_7$ $\varphi \in \{0, 2, 4, 6\}$. From here $\varphi - 3 \in \{-3; -1; 1; 3\}$ and $(\varphi - 3)^2 \in \{9; 1; 1; 9\}$. $(\varphi - 3)^2 - 1 \in \{8; 0; 0; 8\}$ and $f_0 = ((\varphi - 3)^2 - 1)/8 \in \{1; 0; 0; 1\}$. Obviously, the derived function $f_0 = f_0(x_1; x_2; \ldots x_7)$ is a Boolean function, because all of its variables $x_i \in \{0; 1\}$ $f_0 \in \{0; 1\}$. Now let us show that $deg(f_0) = 4$:

$$\varphi = x_1^2 - 2x_1x_2 + x_2^2 - x_2^2 - 2x_2x_3 + x_3^2 + \ldots - x_7^2 - 2x_7x_1 + x_1^2 =$$
$$= 2(x_1 + x_2 + \ldots + x_7) - 2(x_1x_2 + x_2x_3 + \ldots + x_7x_1),$$

therefore $deg(\varphi) = 2$. Expression $(\varphi - 3)^2$ contains all the elements of the set squared:

$$A = \{2x_1; 2x_2; \ldots; 2x_7; -2x_1x_2; -2x_2x_3; \ldots; -2x_7x_1; -3\}$$

the degree of it will be 1 or 2 because $x_i^2 = x_i$. Also the expression $(\varphi - 3)^2$ contains products of any two distinct elements of set A doubled, which forms set $B = \{8x_1x_2; 8x_2x_3; \ldots; 8x_7x_1; 8x_1x_2x_3x_4;$ etc.$\}$. Set B contains monomials of degree 4, for example, $8x_1x_2x_3x_4$, but no monomial has a degree higher than 4. $(\varphi - 3)^2$ is the sum of such monomials therefore $deg((\varphi - 3)^2) = 4$. Therefore also

$$deg(f_0) = deg(((\varphi - 3)^2 - 1)/8) = 4.$$

Now let us clarify the value of $D(f_0)$. As a result from the above, we see that

$$f_0 = \begin{cases} 0 & \text{if } d = 2 \text{ or } d = 4 \\ 1 & \text{if } d = 0 \text{ or } d = 6 \end{cases}$$

(d is the number of differences).

If we know any 6 values of the variables and they all are equal then we can not calculate the value of f_0. Really, if the 7th variable is equal to other 6 variables then $d = 0$ and $f_0 = 1$, but if the variable is different then $d = 2$ and $f_0 = 0$. Therefore $D(f_0) > 6$ and $D(f_0) = 7$. Therefore also $f_0 \in B\langle 4; 7; 7\rangle$.

In [1] Ambainis & Freivalds proposed function $f \in B\langle 4; 7; 7\rangle$, which we denote with f_{AF}. Let us show that functions f_0 and f_{AF} are different.

$\sum_{i=1}^{7} x_i$	Author's function $f_0 = f_0(x_1; x_2; \ldots; x_7)$	Ambainis & Freivalds function $f_{AF} = f_{AF}(x_1; x_2; \ldots; x_7)$
0	$f_0 = 1$	$f_{AF} = 0$
1	$f_0 = 0$	$f_{AF} = 1$
2	$f_0 = 0$	$f_{AF} = 1$
3	$f_0 = 0$ or $f_0 = 1$	$f_{AF} = 0$ or $f_{AF} = 1$
4	$f_0 = 0$ or $f_0 = 1$	$f_{AF} = 0$ or $f_{AF} = 1$
5	$f_0 = 0$	$f_{AF} = 0$
6	$f_0 = 0$	$f_{AF} = 0$
7	$f_0 = 1$	$f_{AF} = 0$

Table 1

From Table 1 we see that $f_0 \not\equiv f_{AF}$ because if the sum $\sum_{i=1}^{7} x_i$ is 0, 1 or 2, then

$$f_0(x_1; x_2; \ldots; x_7) \neq f_{AF}(x_1; x_2; \ldots; x_7).$$

Also we see that $f_0 \not\equiv 1 - f_{AF} = \overline{f_{AF}}$ because if sum $\sum_{i=1}^{7} x_i$ is 5 or 6, then $f_0 = f_{AF}$. Also we note that for all x_i

$$f_0(x_1; x_2; \ldots; x_7) \equiv f_0(\overline{x}_1; \overline{x}_2; \ldots; \overline{x}_7), \text{ but}$$

$$f_{AF}(x_1; x_2; \ldots; x_7) \not\equiv f_{AF}(\overline{x}_1; \overline{x}_2; \ldots; \overline{x}_7) \ (\overline{x}_i = 1 - x_i).$$

From here we conclude that set $B\langle 4; 7; 7\rangle$ contains at least two functions.

3. Graph - polynomial method.

Now reconsider how we got the function f_0. Obviously the technique written above can be generalize, creating an algorithm for constructing functions $f(x_1; x_2; \ldots; x_7)$:

1. Let us take any Boolean function $h(x; y)$ with two variables which has no fictive variables (for example, in case of function f_0 we consider function $h(x; y) = (x - y)^2 = x + y - 2xy$).

2. Let us construct some graph G with n vertices. For each vertice we assign a variable x_i. After then for each graph edge $(x_i; x_j)$ we calculate $h(x_i; x_j)$ and the sum for all edges. Then we get a function $\varphi(x_1; x_2; \ldots; x_n)$:

$$\varphi(x_1; x_2; \ldots; x_n) = \sum_{\substack{1 \le i < j \le n \\ (x_i; x_j) \in G}} h(x_i; x_j),$$

where $(x_i; x_j) \in G$ means that the graph G contains an edge $(x_i; x_j)$.

3. Now we clarify what values can the function φ assume depending from the variables. Then we establish that for all x_i $\varphi(x_1; x_2; \ldots; x_n) \in A$, where $A = \{a_1; a_2; \ldots; a_k\}$, $a_i \in \mathbf{N} \cup \{0\}$.

4. Let us construct such polynom $P = P(y)$ that for $\forall\, y \in A$ $P(y) \in \{0; 1\}$. We will try to minimalize the degree of the polynom.

5. Let us consider function $f(x_1; x_2; \ldots; x_n) = P(\varphi(x_1; x_2; \ldots; x_n))$. This function is a Boolean function with $deg(f) \le deg(P) \cdot deg(\varphi) = 2 \cdot deg(P) \le 2 \cdot (|A| - 1) = 2 \cdot (k - 1)$. If we manage to prove that $D(f)$ is sufficiently large or even $D(f) = n$ and k is small, then the function f is suitable.

We will call this technique for constructing such functions the **graph – polynomial method**. For $h(x; y) = (x - y)^2$ this method allows us to construct function f which the resulting $deg(f)$ is an even number. Also for these functions for all x_i $f(x_1; x_2; \ldots; x_n) = f(\overline{x}_1; \overline{x}_2; \ldots; \overline{x}_n)$.

To use this method further, we will make several definitions and prove some lemmas.

4. Notations, definitions, lemmas and theorems.

Definition 1. A n – cyclic graph ($n \ge 3$) is a graph with edges $(a_1; a_2)$, $(a_2; a_3)$, $\ldots$, $(a_{n-1}; a_n)$, $(a_n; a_1)$. We will denote such graph with C_n (Figure 3).

Definition 2. If i th vertice has a variable $x_i \in \{0; 1\}$, then difference is an edge where the vertices of that edge have different values. The difference count for this graph will be the total sum of differences.

Lemma 1. For every $n \ge 3$ the number of differences in graph C_n will always be an even number.

Proof. We assign variable x_i to vertice i. The sum

$$(x_1 - x_2) + (x_2 - x_3) + \ldots + (x_{n-1} - x_n) + (x_n - x_1) = 0$$

is an even number. For edges which have equal values on the corresponding vertices, the difference is 0, for edges having different values on corresponding vertices, the difference is an odd number. Therefore the sum of odd numbers will be an even number, as the number of odd numbers is equal to the difference count. Therefore the number of odd numbers and the difference count can only be even.

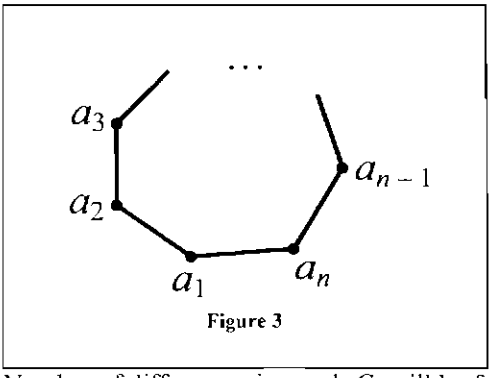


Figure 3

Conclusion. Number of differences in graph C_n will be from set $\{0; 2; 4; \ldots; 2\cdot\left\lfloor\frac{n}{2}\right\rfloor\}$. It can be verified that the number of differences can take any of those values, therefore none of them can be discarded. We omit this proof.

Definition 3. Multilinear polynomial is a polinomial which can be expressed in form

$$p(x_1; x_2; \ldots; x_n) = a_0(1) + a_1(1)x_1 + a_1(2)x_2 - \ldots + a_1(n)x_n + a_2(1)x_1x_2 + a_2(2)x_2x_3 + \\ + \ldots + a_2(C_n^2)x_{n-1}x_n + \ldots + a_3(1)x_1x_2x_3 + \ldots + a_n(1)x_1x_2 \ldots x_n.$$

Lemma 2. If f is multilinear polynomial with n variables which all from set $\{0; 1\}$ and $deg(f) = k$, $k \le n$ then f^m is also multilinear polynomial and

$$deg(f^m) \le min(k\cdot m, n).$$

(we omit the proof)

Definition 4. For function f with an input vector $x = (x_1; x_2; \ldots; x_n)$ sensitivity is the number of arguments x_i which have to be changed in order to change the value of the function:

$$f(x_1; \ldots; x_i; \ldots; x_n) \ne f(x_1; \ldots; 1 - x_i; \ldots; x_n).$$

We will denote it with $s_x(f)$. The maximum of $s_x(f)$ for all vectors x will be denoted as $s(f)$.

Lemma 3. (Buhrman and de Wolf, 2001) For every Boolean function f $s(f) \le D(f)$.

Lemma 4. If there exists a vector $x = (x_1; x_2; \ldots; x_n)$ such that $s_x(f) = n$ then function f does not have any fictive variables.

Proof. If any of the variables x_i is changed, the value of the function changes. therefore none of the variables is fictive.

5. Construction of functions having $deg(f) < D(f)$.

Using the obtained graph - polynomial method, proved lemmas and introduced notations we will construct some functions.

Theorem 1. There is a function $f \in B\langle 2; 2; 2\rangle$.

Proof. Let us consider function $h(x_1; x_2) = (x_1 - x_2)^2 = x_1 + x_2 - 2x_1x_2$. We can see that for every x_1, x_2 $h \in \{0; 1\}$ and $deg(h) = 2$. As $h = 0$ if $x_1 = x_2$ and $h = 1$ if $x_1 \ne x_2$ (difference) then $s_{(0; 0)}(f) = 2$ and $D(h) = 2$.

Theorem 2. There is a function $f \in B\langle 2; 3; 3\rangle$.

Proof. Let us consider graph C_3. Number of differences for it can be 0 or 2. Therefore $\varphi(x_1; x_2; x_3) \in \{0; 2\}$, where $\varphi(x_1; x_2; x_3) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_1)^2$. Therefore $f = \varphi/2 \in \{0; 1\}$ and

$$f = x_1 + x_2 + x_3 - x_1x_2 - x_1x_3 - x_2x_3.$$

$deg(f) = 2$ and $D(f) = 3$ because $s_{(0; 0; 0)}(f) = 3$. Therefore $f \in B\langle 2; 3; 3\rangle$.

Remark. As we can see, the obtained function is the same Nisan function mentioned above. Consequently this function can be also interpretated like the function from the number of differences.

Theorem 3. There is a function $f \in B\langle 4; 4; 4\rangle$.

Proof. Let us consider graph C_4. The difference count can be either 0, 2 or 4. Therefore function

$$\varphi(x_1; x_2; x_3; x_4) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_1)^2$$

will take values from set $\{0; 2; 4\}$. Further, $(\varphi - 2)^2 \in \{4; 0; 4\}$ and $f = (\varphi/2 - 1)^2 \in \{1; 0; 1\} = \{0; 1\}$. We see that $deg(f) = 4$ and $D(f) = 4$ because $s(f) = 4$. Therefore $f \in B\langle 4; 4; 4\rangle$.

Theorem 4. There is a function $f \in B\langle 4; 5; 5\rangle$.

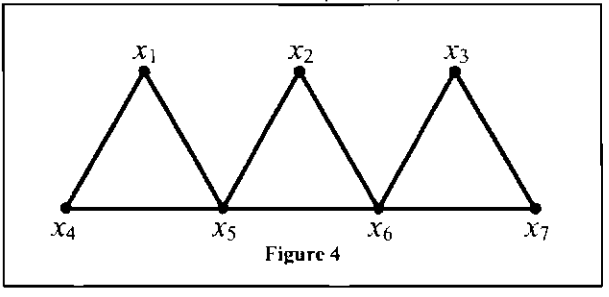


Figure 4

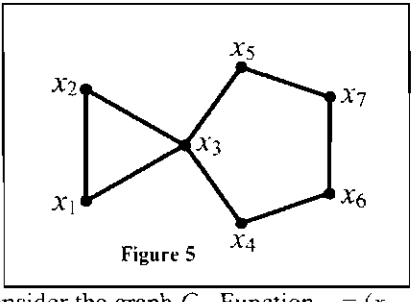


Figure 5

Proof. Let us consider the graph C_5. Function $\varphi = (x_1 - x_2)^2 + (x_2 - x_3)^2 + \ldots + (x_5 - x_1)^2$ which is equal to diference count will take value from set $\{0; 2; 4\}$, therefore $f = (\varphi/2 - 1)^2 \in \{1; 0; 1\} = \{0; 1\}$. $deg(f) = 4$ and $D(f) = 5$ because $s_{(0, 0, 0, 0, 0)}(f) = 5$. Therefore $f \in B\langle 4; 5; 5\rangle$.

Remark. By using the graph C_6, we can obtain function $f \in B\langle 4; 6; 6\rangle$, which is less "interesting" than Kushilevitz function $f \in B\langle 3; 6; 6\rangle$.

Theorem 5. There are different functions $f_1, f_2 \in B\langle 4; 7; 7\rangle$, which differs from the previously considered 7 – argument functions f_0 and f_{4F}.

Proof. Let us consider graphs seen in Figure 4 and Figure 5. The total number of differences can be either 0, 2, 4 or 6, as every triangle in the graphs can have a

value from set {0; 2} and pentagon in Figure 5 can have a value from set {0; 2; 4}. Therefore $\varphi_1, \varphi_2 \in \{0; 2; 4; 6\}$, where

$\varphi_1 = (x_4 - x_1)^2 + (x_4 - x_5)^2 + (x_1 - x_5)^2 + \ldots + (x_6 - x_3)^2 + (x_6 - x_7)^2 + (x_3 - x_7)^2$ and

$\varphi_2 = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_1 - x_3)^2 + (x_3 - x_5)^2 + (x_5 - x_7)^2 + \ldots + (x_4 - x_3)^2$.

Then $f_1 = ((\varphi_1 - 3)^2 - 1)/8$, $f_2 = ((\varphi_2 - 3)^2 - 1)/8$. $f_1, f_2 \in \{1; 0; 0; 1\} = \{0; 1\}$. $deg(f_1) = deg(f_2) = 4$ and $D(f_1) = D(f_2) = 7$, because $s_{(0; 0; \ldots; 0)}(f_1) = s_{(0; 0; \ldots; 0)}(f_2) = 7$.

Now we can compare the functions f_0, f_1, f_2 and f_{AF}, which comes from set $B\langle 4; 7; 7\rangle$ (Table 2).

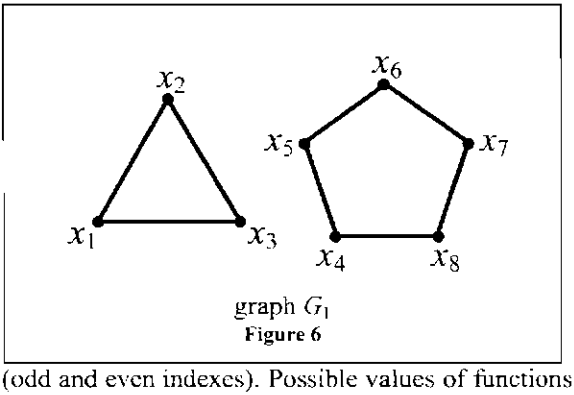
sum of x_i	0	1	2	3	4	5	6	7
f_0	1	0	0	0/1	0/1	0	0	1
f_1	1	0	0/1	0/1	0/1	0/1	0	1
f_2	1	0	0/1	0/1	0/1	0/1	0	1
f_{AF}	0	1	1	0/1	0/1	0	0	0

Table 2

From Table 2 we can see that functions f_1 and f_2 are different from all other functions and different are functions f_0 and f_{AF}. There is only question about functions f_1 and f_2. To answer this question we can consider sums $x_1 + x_3 + x_5 + x_7$

even→ odd↓	0	1	2	3
0	1|1	0|0	0,1|0	1|0
1	0|0	0,1|0,1	0,1 0,1	0.1|0,1
2	0,1|0,1	0,1|0,1	0,1 0,1	0,1 0.1
3	0,1|0,1	0,1|0,1	0,1 0,1	0|0
4	1|0	0,1|0	0|0	1|1

Table 3



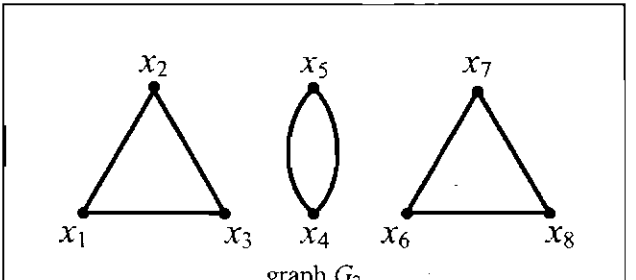
graph G_1

Figure 6

and $x_2 + x_4 + x_6$ (odd and even indexes). Possible values of functions f_1 and f_2 in this case we can see in Table 3 (in every cell in left side is all possible values of function f_1 and in right side is all possible values of function f_2). Author think that from Figure 4, Figure 5 and Table 3 follow that functions f_1 and f_2 are different. Now we know that set $B\langle 4; 7; 7\rangle$ contains at least four different functions.

Theorem 6. There are different functions $g_1, g_2 \in B\langle 4; 8; 8\rangle$.

Proof. To construct 8 – argument functions, we will not use graph C_8, but we will consider graph G_1 and G_2 (Figure 6 and Figure 7). Easy to see that both graphs G_1 and G_2 are disjoint and G_2 is a multigraph.



graph G_2

Figure 7

We can see that difference count of each graph G_1 and G_2 will take this and only this values: 0, 2, 4 and 6. Therefore $\varphi_1, \varphi_2 \in \{0; 2; 4; 6\}$ where φ_1 and φ_2 is difference count function corresponding graphs G_1 and G_2. Further, $g_1 = ((\varphi_1 - 3)^2 - 1)/8 \in \{0; 1\}$ and $g_2 = ((\varphi_2 - 3)^2 - 1)/8 \in \{0; 1\}$. We see that $deg(g_1) = deg(g_2) = 4$ and $D(g_1) = D(g_2) = 8$ because $s_{(0;\,0;\,\ldots;\,0)}(g_1) = s_{(0;\,0;\,\ldots;\,0)}(g_2) = 8$. Therefore $g_1, g_2 \in B\langle 4; 8; 8\rangle$.

Obviously, functions g_1 and g_2 differs each from other. Analysing diference count for several input variables, we get Table 4.

sum of x_i	0	1	2	3	4	5	6	7	8
g_1	1	0	0	0/1	0/1	0/1	0	0	1
g_2	1	0	0/1	0/1	0/1	0/1	0/1	0	1

Table 4

Now we know that set $B\langle 4; 8; 8\rangle$ contains at least two different functions.

References.

[1] Andris Ambainis, Rūsiņš Freivalds. Boolean function with a low polynomial degree. "Latvijas Zinātņu Akadēmijas Vēstis", section B, vol. 57, No. 3/4 (626/627), p. 74-77.

[2] Andris Ambainis: Polynomial Degree vs. Quantum Query Complexity. FOCS 2003: 230-239.

[3] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, Ronald de Wolf: Quantum Lower Bounds by Polynomials. FOCS 1998: 352-361.

[4] Rūsiņš Freivalds, Masahiro Miyakawa, Ivo G. Rosenberg: Complexity of Decision Trees for Boolean Functions. ISMVL 2003: 253-256.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Datbases and Information Systems
40–45 P.

Improved quantum lower bounds for 3-Sum problem

Andrej Dubrovsky and Oksana Scegulnaja-Dubrovska

University of Latvia, IMCS
29 Raina boulevard, Riga, Latvia
andrejs.dubrovskis@alise.lv, oksana.s@lis.lv

Abstract. In this work we observe a fundamental problem in Computational geometry that is called 3-sum. This problem makes up a whole class, every problem of which is reducible to 3-Sum. This class is called 3-Sum Hard. In this paper we improve lower bound $\Omega(\sqrt{N})$ obtained in [8]. Our new result is tight lower bound $\Omega(NlogN)$ and uses the fact, that quantum binary tree search is not much faster than its classical counterpart.

Keywords. Quantum computation, lower bound

1. Introduction

The quantum mechanism gives us a certain kind of power, which cannot be achieved by the deterministic or probabilistic approach. The aim of quantum computation is to discover tasks, where quantum computation is significantly faster than classical one. There are some common methods, that allow to speed up quantum computation. The database search algorithm [11] introduced by Grover gives quadratic speedup comparing to classical search. It has inspired a number of other new quantum algorithms, that use it as a building block. We will mention amplitude amplification algorithm [6], that can speed up almost all quantum algorithms.
On the other hand, not every algorithm can be sped up even by such a powerful method. For example, binary tree search in quantum case has the same bound as its classical counterpart - $\Omega(logN)$ [3]. In this paper we will show the use of this algorithm to find a lower bound for quantum 3-Sum problem.

2. Definitions and previous results

In this paper we observe the following problem, called 3-Sum problem:
Definition 1. Given the set S of N numbers, detect whether there are three numbers $a \in S$; $b \in S$; $c \in S$, such that $a + b + c = 0$.

Alternative model, often called 3-sum', is:
Definition 2. Given the 3 sets A, B and C each of N numbers, detect whether there are three numbers $a \in A$; $b \in B$; $c \in C$, such that $a + b = c$.

There is a big cluster of problems in computational geometry that are called 3-Sum Hard. Gajentaan and Overmas [10] described them as problems that can be reduced to the 3-Sum problem. The example is, for instance, a GeomBase problem: given points on three equally spaced horizontal lines, are there points, one from each line, that are collinear.

In classical computation the best currently known algorithm for any 3-Sum Hard problem takes $O(N^2)$ time, while the best lower bound for the time complexity is Ω (*NlogN*), which is very low and unreachable. It is believed that 3-Sum lower bound is the same as upper bound, $\Omega(N^2)$, so 3-Sum hard problems in classical computation sometimes call - N^2 *hard*. For some of 3-Sum hard problems $\Omega(N^2)$ lower bound has been proved.

Considering quantum case, Dubrovsky and Scegulnaja [8] have shown quantum algorithm, that solves this problem in $O(NLogN)$ time, and lower bound $\Omega(\sqrt{N})$, although very low and unreachable. They also gave lower bound $\Omega(\sqrt{N})$ and algorithm, that works within $O(N^{\lceil r/3 \rceil} \log N)$ for generalization of 3-Sum called r-Sum.

In Section 3 we discuss quantum algorithm for 3-Sum problem [8].

In Section 4 we show how classical lower bound for 3-sum can be proved, using binary search method and use the same method in quantum case. We show that, although this lower bound is not tight in classical case, it appears to be tight enough in quantum case due to the fact, that quantum binary search cannot give much speedup comparing to classical algorithm.

In section 5 we show 3-Sum generalization called r-Sum. We discuss algorithm and new lower bound.

3 Quantum algorithm for the 3-Sum problem

In this section we show algorithm and lower bound, that is presented in [8].

In our algorithm we make use of quantum amplitude amplification method, which generalizes Grover quantum search. Here is an essence of amplitude amplification:

Theorem 1. There exists the quantum algorithm QSearch with the following property. Let A be any quantum algorithm that uses no measurements, and let $\gamma : Z \to \{0,1\}$ be any boolean function. Let a denote the initial success probability of A of finding a solution (i.e. the probability of outputting z such that $\gamma(z) = 1$). Algorithm QSearch finds a solution using an expected number of $O(1/\sqrt{a})$ applications of A and A^{-1} if a > 0, and otherwise runs forever.

The algorithm QSearch does not need to know the value of a in advance, but if a is known, it can find a solution in worst-case $O(1/\sqrt{a})$ applications.

Theorem 2. There exists a quantum algorithm that solves 3-Sum problem in O(NlogN).

Proof. The algorithm works as follows:

1. Sort set C classically, that takes O(NlogN) time.

2. Construct an algorithm that can solve the problem with small probability.

The algorithm takes an input of two random elements, one from set A and the other from set B and outputs whether these two elements are summing up to some element from C:

(a) Compute a+b, $a \in A$; $b \in B$.

(b) Check whether a+b can be found in C. Needs $O(logN)$ time, because C is sorted.

3. Construct quantum superposition over all the $|a\rangle|b\rangle$ and use amplitude amplification on that superposition with the algorithm just described as a kernel. Amplitude amplification method uses Grover algorithm idea to speed up computation.

The maximum speedup it allows to get is quadratic. In our case, classically we must repeat algorithm kernel steps $O(N^2)$ times. Amplitude amplification method allows us to get the same result, using only O(N) steps.

So the total time the algorithm uses is O(NlogN).

There are several approaches for estimating lower bounds for quantum algorithms.

Most popular are classic adversary method [5],[13], which tries to alternate the input without affecting the output , polynomials [4], which uses polynomials to estimate lower bound of quantum algorithms, and quantum adversary method [1], [2], that runs on superposition of inputs and estimates the number of queries needed to achieve entanglement between algorithm and oracle work spaces. We used Ambainis quantum adversary method to compute lower bound in [8] :

Theorem 3. 3-Sum problem has quantum lower bound $\Omega(\sqrt{N})$

Proof. The proof is based on A. Ambainis adversary method of proving quantum lower bounds [1], [2]. We use his main theorem.

Theorem 4. Let $f(x_1,x_2, \ldots ,x_n)$ be a function of N $\{0, 1\}$ - valued variables and X, Y be two sets of inputs such that $f(x) \neq f(y)$ if $x \in X$ and $y \in Y$. Let $R \subset X^* Y$ be such that

For every $x \in X$ there exist at least m different $y \in Y$ such that $(x, y) \in R$,

For every $y \in Y$ there exist at least m' different $x \in X$ such that $(x, y) \in R$,

For every $x \in X$ and $i \in \{1, \ldots, n\}$ there are at most l_i different $y \in Y$ such that $(x, y) \in R$ and $x_i \neq y_i$.

For every $y \in Y$ and $i \in \{1, \ldots, n\}$ there are at most l_i' different $x \in X$ such that $(x, y) \in R$ and $x_i \neq y_i$,

Then, any quantum algorithm computing f uses $\Omega\left(\sqrt{\frac{mm'}{\max(l_i * l'_i)}}\right)$ queries.

That means, that we need to find all the variants how the input x can be modified. For our case, we'll take X to contain only one element: A consisting of all zeros; B -

all 1s, C - all 2s. It is an input, on which our function returns 0. Let Y contain all inputs made of X, with one element in any of sets A, B and C changed so, that the function returns 1 (e.g., any element of A changed to 1; any element of B - to 2; any element of C - to 1). Let R consist of such $X * Y$ pairs where y differs from x in exactly one position. According to the theorem, $m = 3N$, because for every $x \in X$ there are exactly 3N different $y \in Y$, which differs from x in exactly one position. $m' = 1$, because for every $y \in Y$ there is only one $x \in X$, which differs from y in exactly one position. $l = l' = 1$ that follows from our definition of R.

Using this formula we get a lower bound $\Omega(\sqrt{N})$.

Unfortunately, this method gives almost trivial result in this case. The better idea was to try the same method as in classical case.

4 Improved lower bound for 3-Sum problem

Classical lower bound $\Omega(NlogN)$ follows from the technique of Dobkin and Lipton [7] in the linear decision tree model. They observed that the set of inputs following a fixed computational path through a linear decision tree is connected. Since the set of nondegenerate inputs has $n^{\Omega(n)}$ connected components, any linear decision tree must have $n^{\Omega(n)}$ leaves and therefore must have depth Ω ($NlogN$). As quantum algorithm cannot give any speedup on a linear decision tree, we must conclude, that quantum lower bound for 3-Sum is Ω ($NlogN$).

5 3-sum generalization

The most natural generalization of 3-Sum problem is its r-Sum, as defined in [9]:
Definition 3. Given a set S of N numbers, detect whether there are r numbers in S which sum to zero.
Alternative definition called r-Sum' is the following:
Definition 4. Given r sets $S_1,\ldots, S_r$ of N numbers, detect whether there are r numbers one from each set that sum to zero.

Similarly we can define the class of r-Sum Hard problems.
In deterministic case these problems have a lower bound of Ω ($Nlog(N)$) and best known deterministic algorithm can solve the r-sum problem in $O(N^{r+1/2})$ when r is odd and $O(N^{r/2}\log(r))$ when r is even.
Theorem 5. Quantum algorithm described in [8] can solve r-Sum problem in $O(N^{\lceil r/3 \rceil}\log N)$ time.
Proof. We will further divide sets $S_1, \ldots S_r$ in two more groups of sets: First group will contain x sets of data - let's call them $C_1,\ldots,C_x$ and the second will contain remaining (r-x) sets of data - let's call them $Q_1,\ldots,Q_{r-x}$.
The algorithm itself consists of two parts - classical and the quantum one.

First, we execute the classical part of the algorithm and then execute the quantum part of it.

Classical part of algorithm works as follows: we take sets $C_1,...,C_x$ and perform following operations on them:

1. Make all the possible groups of elements picking one from each of sets $C_1,...,C_x$ and sum them up. (Call this new set CSum)
2. Sort the summary set CSum

After classical part of algorithm finishes its work we will get the sorted set of all the possible element combinations in sets $C_1,...,C_x$. This will take $O(N^x \log N)$ time.

Then starts the quantum part of the algorithm, that uses sets $Q_1,...,Q_{r-x}$ as well as the set CSum. The quantum part of r-sum algorithm is in fact generalized version of quantum 3-sum algorithm:

1. Construct the probabilistic algorithm that can find the solution of the problem with a small probability. The algorithm will take two steps: This algorithm will randomly take one element from each of sets $Q_1,...,Q_{r-x}$ and sum those elements. Then it will take the sum obtained and search for it in CSum sorted set.

The first algorithm step can be accomplished in constant time that depends only on r. The second algorithm part takes $O(log(N))$ time to search element in sorted database. So the total running time of the algorithm is $O(log(N))$. However this algorithm will find the solution with probability only $\frac{1}{N^{r-x}}$

2. To boost the probability of success we use amplitude amplification technique. We prepare the starting quantum superposition $\sum |Q_1\rangle |Q_2\rangle ... |Q_{r-x}\rangle$ and call the amplitude amplification with algorithm from the step 1 embedded in it. Then, after $O(N^{(r-x)/2})$ steps the algorithm will give an answer with high probability of success. So the running time of quantum part of algorithm is $O(N^{(r-x)/2} \log N)$.

The total running time of this algorithm is: $O(\max(N^x \log N, N^{(r-x)/2} \log N))$. so we should minimize this function. This is the case when x = (r-x)/2. So we get the value of x = 1/3r and total algorithm running time $O(N^{\lceil r/3 \rceil} \log N)$.

Theorem 6. r-Sum problem has quantum lower bound Ω (NlogN)

Proof. The proof is the same as for 3-Sum problem.

6 Conclusion

In this paper we have observed 3-Sum problem. This problem is of great importance in computational geometry, because there exists a whole class of related problems called 3-Sum Hard that can be reduced to 3-Sum. We have shown quantum algorithms from 3-Sum problem and its generalization called r-Sum, that run faster that their classical counterparts. As to lower bounds, we have improved our previous results, finding a tight lower bound for 3-Sum problem and better lower bound for r-Sum problem.

References

[1]. Ambainis, A.: Quantum lower bounds by quantum arguments. Manuscript, 1999.
[2]. Ambainis, A.: Quantum query model: algorithms and lower bounds.Proceedings of the Second International Workshop on Quantum Computing and Learning, 1999.
[3]. Ambainis, A.: A better lower bound for quantum algorithms searching an ordered list. arXiv:quant-ph/9902053 v1
[4]. Beals, R., Buhrman, H., Cleve, R., Mosca, M., de Wolf, R.: Quantum lower bounds by polynomials. Proceedings of FOCS'98, pages 351–361. Also arXiv:quantph/9802049v3, 1998.
[5]. Bennett, C.; Bernstein, E.; Brassard, c.; Vasirani, U.: Strengths and weaknesses of quantum computing. SIAM Journal on Computing, 26(3):1510-1523, 1997, quantph/9701001.
[6]. Brassard, G.; Hoyer, P.; Tapp, A.: Quantum algorithm for the collision problem. ACM SIGACT News (Criptology column), 28:14-19, 1997. quant-ph/9705002.
[7]. Dobkin, D.P., Lipton, R.J.: On the complexity of computations under varying sets of primitives. J. Comput. Syst. Sci.,18:86-91, 1979.
[8]. Scegulnaja, O, Dubrovsky, A.: Quantum algorithms and lower bounds for 3-Sum problem. In proceedings of EQIS'2003, pages 131-132, 2003.
[9]. Erickson, J.: New lower bound for convex hull problems in odd dimensions. In Proceedings of the twelfth annual symposium on Computational geometry, pages 1-9. , ACM Press, 1996.
[10]. Gajentaan, A.; Overmars, M. H.: On a class of $o(n^2)$ problems in computational geometry. Computational Geometry: Theory and Applications, 5(3):165-185,1995.
[11]. Grover, L. K.: A fast quantum mechanical algorithm for database search. In proceedings of 28th STOC, pages 212-219, 1996.
[12]. Gruska, J.: Quantum computing, McGraw Hill (1999).
[13]. Shi, Y.: Lower bounds of quantum black - box complexity and degree of approximation polynomials by influence of Boolean variables. Information Processing Letters, 75:79-83, 2000, quant-ph/9904107.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004, VOLUME 673, Databases and Information Systems
46-56 P.

Lower Bounds for Query Complexity of Some Graph and Matrix Problems

Lelde Lāce and Rūsiņš Freivalds[1]

Institute of Mathematics and Computer Science University of Latvia
Raiņa bulvāris 29, Riga, LV-1459, Latvia
E-mail addresses: (Lelde.Lace, Rusins.Freivalds)@mii.lu.lv

Abstract The paper [4] by H.Buhrman and R. de Wolf contains an impressive survey of solved and open problems in quantum query complexity, including many graph problems. We use recent results by A.Ambainis [1] to prove higher lower bounds for some of these problems. Some of our new lower bounds do not close the gap between the best upper and lower bounds. We prove in these cases that it is impossible to provide a better application of Ambainis' technique for these problems.

Keywords. Quantum algorithm, lower bound, graph-theory

1. Introduction

Recently it has become clear that a quantum computer could, in principle, solve certain problems faster than a conventional computer. A quantum computer is a device, which takes full advantage of quantum mechanical superposition and interference. Building an actual quantum computer is probably far off in the future. Boolean decision trees model is the simplest model to compute Boolean functions. In this model the primitive operation made by an algorithm is evaluating an input Boolean variable. The cost of a (deterministic) algorithm is the number of variables it evaluates on a worst-case input.

The *black-box* model of computation arises when one is given a black-box containing an N-tuple of Boolean variables $X=(x_1,x_2,....x_N)$. The box is equipped to output x_i on input i. We wish to determine some property of X, accessing the x_i only through the black-box. Such a black-box access is called a *query*. A property of X is any Boolean function that depends on X, i.e. a property is function $f:\{0,1\}^N \rightarrow \{0,1\}$. We want to compute such properties using as few queries as possible.

Consider, for example, the case where the goal is to determine whether or not X contains at least one 1, so we want to compute the property $OR(X) = x_0 \vee x_1 ... \vee x_{N-1}$. It is well known that the number of queries required to compute OR by any *classical* (deterministic or probabilistic) algorithm is O(N).Grover [6] discovered a

[1] Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP)

remarkable *quantum* algorithm that, making queries in superposition, can be used to compute OR with small error probability using only $O(\sqrt{N})$ queries.

On the other hand, quantum algorithms are in a sense more restricted. For instance, only unitary transformations are allowed for state transitions. Hence rather often a problem arises whether or not the needed quantum automaton exists. In such a situation lower bounds of complexity are considered. It is proved in [3] that Grover database search algorithm is the best possible. It is proved in [3] that no quantum query algorithm exists for PARITY with $\Omega(N)$ queries, etc.

We use a result by A.Ambainis [1] to prove lower complexity bounds for quantum query algorithms. Currently, this is the most powerful method to prove lower bounds of complexity for quantum query algorithms. In some cases there still remains a gap between the upper and the lower bounds of the complexity. In these cases we prove additionally that Ambainis' method cannot provide a better lower bound for this problem.

2. Query model

In the query model, the input $x_1,...x_N$ is contained in a black box and can be accessed by queries to the black box. In each query, we give i to the black box and the black box outputs x_i. The goal is to solve the problem with the minimum number of queries. The classical version of this model is known as *decision trees* [4].

There are two ways how to define the query box in the quantum model. The first is the extension of the classical query (Figure 1).

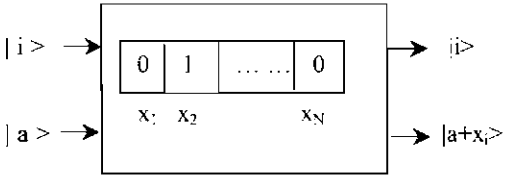


Figure1. Quantum black box.

It has two inputs i, consisting of [logN] bits and b consisting of 1 bit. If the input to the query box is a basic state $|i\rangle|b\rangle$, the output is $|i\rangle|b\oplus x_i\rangle$. If the input is a superposition $\sum_{i,b} a_{i,b}|i\rangle|b\rangle$, the output is $\sum_{i,b} a_{i,b}|i\rangle|b\oplus x_i\rangle$. Notice that this definition applies both to case when x_i are binary and to the case when they are k-valued. In the k-valued case, we just make b to consist of $\lceil \log_2 k \rceil$ bits and take $b\oplus x_i$ to be bitwise XOR of b and x_i.

In the second form of quantum query (which only applies to problems with {0,1}-valued x_i), the black box has just one input i. If the input is a state $\sum_i a_i|i\rangle$, the output is $\sum_i (-1)^{x_i} a_i|i\rangle$. While this form is less intuitive, it is very convenient for the use in quantum algorithms, including Grover's search algorithm [6]. A query of second type can be simulated by a query of first type [6].

A quantum query algorithm with T queries is just a sequence of unitary transformations

$$U_0 \rightarrow O \rightarrow U_1 \rightarrow O \rightarrow \ldots \rightarrow U_{T-1} \rightarrow O \rightarrow U_T$$

on some finite- dimensional space C^k. U_0, U_1, ..., U_T can be any unitary transformations that do not depend on the bits x_1, ...,x_N inside the black box. O are query transformations that consist of applying the query box to the first logN+1 bits of the state. That is, we represent basic states of C^k as $|i,b,z\rangle$. Then, O maps $|i,b,z\rangle$ to $|i,b\oplus x_i,z\rangle$. We use O_x to denote the query transformation corresponding to an input $x=(x_1,\ldots x_N)$.

The computation starts with state $|0\rangle$. Then, we apply U_0, O_x ...., O_x, U_T and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement (or several bits if we are considering a problem where the answer has more than 2 values).

The quantum algorithm computes a function $f(x_1,\ldots,x_N)$ if, for every $x=(x_1,\ldots,x_N)$ for which f is defined, the probability that the rightmost bit of $U_T O_x U_{T-1} \ldots O_x U_0|0\rangle$ equals $f(x_1,\ldots,x_N)$ is at least $1-\varepsilon < \frac{1}{2}$.

The query complexity of f is the smallest number of queries used by a quantum algorithm that computes f. We denote it $Q(f)$.

Our proofs use the following results by A.Ambainis.

Theorem A1 [1] *Let $f(x_1, x_2..x_n)$ be a function of n {0,1}-valued variables and $A \subset \{0,1\}^n$, $B \subset \{0,1\}^n$ be such that $f(A)=1$, $f(B)=0$ and*

- *for every $x=(x_1..x_n) \in A$, there are at least m values $i \in \{1,\ldots,n\}$ such that $(x_1, \ldots, x_{i-1}, 1-x_i, x_{i-1}, \ldots, x_n) \in B$,*
- *for every $x=(x_1..x_n) \in B$, there are at least m' values $i \in \{1,\ldots,n\}$ such that $(x_1,\ldots,x_{i-1}, 1-x_i, x_{i-1},\ldots,x_n) \in A$.*

Then $Q(f) = \Omega(\sqrt{mm'})$.

Theorem A2 [1] *Let $f(x_1, x_2..x_n)$ be a function of n {0,1}-valued variables and A, B be two sets of inputs such that $f(x) \neq f(y)$ if $x \in A$ and $y \in B$. Let $R \subset A \bullet B$ be such that*

- *for every $x \in A$ there exist at least m different $y \in B$ such that $(x,y) \in R$,*
- *for every $y \in B$ there exist at least m' different $x \in A$ such that $(x,y) \in R$,*
- *for every $x \in A$ and $i \in \{1,\ldots n\}$ there are at most l different $y \in B$ such that $(x,y) \in R$ and $x_i \neq y_i$,*
- *for every $y \in B$ and $i \in \{1,\ldots n\}$ there are at most l' different $x \in A$ such that $(x,y) \in R$ and $x_i \neq y_i$.*

Then $$Q(f) = \sqrt{\frac{mm'}{\max(ll')}}$$

Definition *For any Boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$ and any $x=(x_1..x_n)$, $ND(f,x)$ is the number of queries needed by nondeterministic algorithms on the values $x=(x_1..x_n)$.*

Definition *For any Boolean function* $f:\{0,1\}^N \to \{0,1\}$
$ND_0(f) = \max_{f(x)=0} ND(f,x)$ *and* $ND_1(f) = \max_{f(x)=1} ND(f,x)$.

Theorem A3 [2] *Whatever the sets A and B, Theorem 1 cannot prove a better lower bound for the query complexity Q(f) than* $\sqrt{ND_0(f) \cdot ND_1(f)}$

We consider the following problems in our paper.

3. Problems

Problem 1 Partition into cliques
Instance: Graph G=(V,E), with |V|=qk for fixed integer q>1 and some integer k.
Question: Can the vertices of G be partitioned into k disjoints sets $V_1,V_2,..V_k$ such that, for $1 \le i \le k$, the subgraph induced by V_i is a complete graph and $|V_i|=q$?

Problem 2 Partition into triangles
Instance: Graph G=(V,E), with |V|=3k for some integer k.
Question: Can the vertices of G be partitioned into k disjoint sets $V_1,V_2, .. V_k$, each containing exactly 3 vertices, such that for each $V_i=\{u_i,v_i,w_i\}$, $1 \le i \le k$, all three of the edges $\{u_i,v_i\},\{u_i,w_i\},\{v_i,w_i\}$ belong to E?

Problem 3 Matching
Instance: Graph G=(V,E), |V|=n.
Question: Can the vertices of G be partitioned into n/2 disjoints pairs $P_1,P_2,....P_{n/2}$ such that for each $P_i=\{u_i,v_i\}$, $1 \le i \le n/2$, edge $\{u_i,v_i\}$ belong to E?

Problem 4 Parity
Instance: Matrix M $2n \times 2n$, $M_{ij} \in \{0,1\}$.
Question: $\sum_{i=1..n} \text{PARITY}(M_i) = n$?

Problem 5 Hamiltonian circuit
Instance: Graph G=(V,E).
Question: Does G contain Hamiltonian circuit?

Problem 6 Vertex cover
Instance: Graph G=(V,E), positive integer $k \le |V|$.
Question: Is there a vertex cover of size k or less for G, i.e. a subset $V' \subseteq V$ with $|V'| \le k$ such that for each edge $\{u,v\} \in E$ at least one of vertices u and v belongs to V?

Problem 7 Dominating set
Instance: Graph G=(V,E), positive integer $k \le |V|$.
Question: Is there a dominating set of size k or less for G, i.e. a subset $V' \subseteq V$ with $|V'| \le k$ such that for all $u \in V-V'$ there is a $v \in V'$ for which $\{u,v\} \in E$?

Problem 8 Chromatic number
Instance: Graph G=(V,E), positive integer $k \leq |V|$.
Question: Does there exist a function $f: V \rightarrow \{1,2,\ldots k\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$?

Problem 9 Monochromatic triangle
Instance: Graph G=(V,E).
Question: Is there a partition of E into two disjoint sets E_1, E_2 such that neither $G1=(V,E_1)$ nor $G2=(V,E_2)$ contains a triangle?

4. Main results

4.1. Partition into cliques

Lemma L1 *If there are k+1 mutually not connected vertices in the graph G=(V,E), |V|=kq,* ***then Partition into cliques*** *problem is not solvable.*
Proof: If there is a solution for Partition into cliques, we get k disjoints sets. Since there is k+1 mutually not connected vertices, there is at least one subset containing two mutually not connected vertices and **Partition into cliques** problem is not solvable. □

Lemma L2 *If a graph G=(V,E), |V|=kq, satisfies the following requirements:*
- *there are k/2 mutually not connected (red) vertices,*
- *there are k green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,*
- *subgraph induced by all the rest vertices (black) is a complete graph and all black vertices are connected to all red and green vertices,*

then ***Partition into cliques*** *problem is solvable.*
Proof: Vertices are grouped in subsets in accordance with the following:
- each red vertex is put in a separate subset (k/2 subsets),
- each pair of green vertices is put in a separate subset (k/2 subsets),
- black vertices are added as follows: q-1 to red and q-2 to green vertices.

Such a distribution satisfies **Partition into cliques** problem. □

Lemma L3 *If graph G=(V,E), |V|=kq, satisfies the following requirements:*
- *there are k/2+2 mutually not connected (red) vertices,*
- *there are k-2 green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,*
- *subgraph induced by all the rest vertices (black) is a complete graph and all black vertices are connected to all red and green vertices,*

then ***Partition into cliques*** *problem is not solvable.*
Proof: If we take red vertices and one from each pair of green vertices then we get k/2+2+(k-2)/2=k+1 vertices. These vertices are not mutually connected. The **Partition into cliques** problem is not solvable because the set of selected vertices satisfies the requirements of Lemma L1. □

Theorem T1 ***Partition into cliques*** *requires* $\Omega(n^{1.5})$ *quantum queries.*
Proof: We construct the sets A and B for the usage of Theorem A1.

The set A consists of all graphs G satisfying the requirements of Lemma L2. The set B consists of all graphs G satisfying the requirements of Lemma L3.

From each graph $G \in A$, we can obtain $G' \in B$ by disconnecting any one of the edges, which connect the green vertices. Hence $m=k/2=O(k)$. From each graph $G \in B$, we can obtain $G' \in A$ by connecting any two red vertices. Hence $m'=(k/2+2)(k/2+1)/2=O(k^2)$.

Since q is fixed, it follows that $k=O(n)$. By Theorem A1, the quantum query complexity is $\Omega\sqrt{n \cdot n^2} = \Omega(n^{1.5})$. □

The same idea proves the following two theorems.

Theorem T2 ***Matching*** *requires* $\Omega(n^{1.5})$ *quantum queries.*

Theorem T3 ***Partition into triangles*** *requires* $\Omega(n^{1.5})$ *quantum queries.*

Theorem T4 *The lower bound for* ***Partition into cliques*** *cannot be improved by Ambainis' method.*
Proof: We use Theorem A3. Let the Boolean function f describe **Partition into cliques.**
$ND_1(f) = O(n)$, because it suffices to ask the edges for all the guessed subsets of vertices; all the subsets are of constant size.
$ND_0(f) = O(n^2)$, because it suffices to exhibit a subset of k−1 vertices connected by no edges. Since $k=O(n)$, $(k-1)k/2 = O(n^2)$.
Hence $\sqrt{ND_1(f) \cdot ND_0(f)} = O(n^{1.5})$. □

4.2. Parity problem

Theorem T5 ***Parity*** *problem requires* $\Omega(n^2)$ *quantum queries.*
Proof: We construct the sets A and B for the usage of Theorem A1.

The set A consists of all matrices M with n rows containing n symbols "1" per row plus n rows containing n−1 symbols "1" per row. The set B consists of all matrices M' with n-1 rows containing n symbols "1" per row plus n+1 rows containing n+1 symbols "1" per row.

Every matrix $M \in A$ can be transformed into a matrix $M' \in B$ by taking an arbitrary row with n symbols "1" and transforming an arbitrary "0" into "1". Hence $m = n^2$. Every matrix $M' \in B$ can be transformed into a matrix $M \in A$ by taking an arbitrary row with n+1 symbols "1" and transforming an arbitrary symbol "1" into "0". Hence $m' = n^2$.

By Theorem A1, the quantum query complexity is $\Omega\sqrt{n^2 \cdot n^2} = \Omega(n^2)$. This is the maximum possible lower bound. □

4.3. Hamiltonian circuit problem

Lemma L4 *If a graph $G=(V,E)$, $|V|=5n$, satisfies the following requirements:*

- *there are n mutually not connected (red) vertices,*
- *there are 2n green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,*
- *all rest 2n vertices (black) are connected to all red and green vertices,*

*then **Hamiltonian circuit problem** is solvable.*

Proof: We make sequence as follows: one red vertex, one black, pair of green, one black, one red, etc. This sequence satisfies **Hamiltonian circuit problem**. □

Lemma L5 *If a graph $G=(V,E)$, $|V|=5n$, satisfies the following requirements:*

- *there are n+2 mutually not connected (red) vertices,*
- *there are 2n-2 green vertices not connected with red ones, green vertices are grouped in pairs and each pair is connected by edge,*
- *all rest 2n vertices (black) are connected to all red and green vertices,*

*then **Hamiltonian circuit problem** is not solvable.*

Proof: The red vertices and the pairs or green vertices are mutually not connected. The only way to get from one red vertex to another (or from one green pair to another) is through some black vertex.

There are 2n black vertices in the graph, but n+2 red vertices and n-1 green pair makes altogether 2n+1. So at least one of the black vertices will be used twice, which is not allowed in **Hamiltonian circuit**. □

Theorem T6 ***Hamiltonian circuit*** *problem requires $\Omega(n^{1.5})$ quantum queries.*

Proof: We construct the sets A and B for the usage of Theorem A1.

The set A consists of all graphs G satisfying the requirements of Lemma L4. The set B consist of all graphs G' satisfying the requirements of Lemma L5.

From each graph $G \in A$, we can obtain $G' \in B$ by disconnecting any one of the edges, which connect the green vertices. Hence $m=n=O(|V|)$. From each graph $G' \in B$, we can obtain $G' \in A$ by connecting any two red vertices. Hence $m'=(n+2)(n+1)/2=O(|V|^2)$.

By Theorem A1, the quantum query complexity is $\Omega\sqrt{n \cdot n^2}=\Omega(n^{1.5})$. □

Theorem T7 *The lower bound for **Hamiltonian circuit** cannot be improved by Ambainis' method.*

Proof: We use Theorem A3. Let the Boolean function f describe **Hamiltonian circuit.**

$ND_1(f) = O(n)$, because it suffices to guess the sequence of vertices and ask the edge for every pair of subsequent vertices.

$ND_0(f) = O(n^2)$, because it suffices to check all edges.

Hence $\sqrt{ND_1(f) \cdot ND_0(f)} = O(n^{1.5})$. □

4.4. Vertex cover problem

Lemma L6 *If a graph G=(V,E), |V|=n, K=n/4, satisfies the following requirements:*

- *there are n/2 black vertices, not mutually connected*
- *there are n/2 red vertices, pairwisely connected,*

then ***Vertex cover*** *problem is solvable.*

Proof: We take in subset one vertex of each red pair and give n/4 vertices, which satisfies **Vertex cover** problem. ⌐

Lemma L7 *If a graph G=(V,E), |V|=n, K=n/4, satisfies the following requirements:*

- *there are n/2-2 black vertices, not mutually connected*
- *there are n/2+2 red vertices, pairwisely connected,*

then ***Vertex cover*** *problem is not solvable.*

Proof: We take in subset one vertex of each red pair and give n/4+1 vertices, which are more than K and doesn't satisfy **Vertex cover** problem. □

Theorem T8 ***Vertex cover*** *problem requires $\Omega(n^{1.5})$ quantum queries.*

Proof: We construct the sets *A* and *B* for the usage of Theorem A1.

The set *A* consists of all graphs *G* satisfying the requirements of Lemma L6. The set *B* consist of all graphs *G'* satisfying the requirements of Lemma L7.

From each graph $G_{\in}A$, we can obtain $G'_{\in}B$ by disconnecting any one of the edges, which connect the red vertices. Hence $m=n/2=O(n)$. From each graph $G'_{\in}B$, we can obtain $G_{\in}A$ by connecting any two black vertices. Hence $m'=(n/2-2)(n/2-3)/2=O(n)$.

By Theorem A1, the quantum query complexity is $\Omega\sqrt{n \cdot n^2}=\Omega(n^{1.5})$. □

The same idea proves the following theorem.

Theorem T9 Dominating set problem requires $\Omega(n^{1.5})$ quantum queries.

4.5. Chromatic number problem

Lemma L8 *If a graph G1 contains one circuit and a graph G2 contains two circuits, then graph G1 and graph G2 separation problem requires $\Omega(n^{1.5})$ quantum queries.*

Proof: We construct the sets *A* and *B* for the usage of Theorem A2.

The set A consists of all graphs G=(V,E), |V|=n, containing Hamiltonian circuit. The set B consists of all graphs G'. Each graph G' is constructed to merge two graphs G"=(V",E"), |V"|=O(n), containing Hamiltonian circuit.

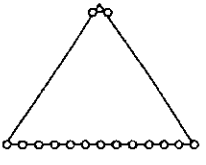


Figure2. Disconnection of Graph G

From each graph $G \in A$, we can obtain $G' \in B$ as follows: disconnecting any one of the edges (n) (Figure 2. top) and one of edges in figure 2. bottom (n/3). Then we make two circuits. From each graph $G' \in B$, we can obtain $G \in A$ as follows: disconnecting one edge in each circuit (n^2) and building up one circuit. Hence $m=n*n/3=O(n^2)$ and $m'=O(n^2)$.

Now we'll find $max(l*l')$. For any edge, which we disconnect in first step $l= n/3$, because we can do that in n/3 combinations with other edges, and $l'=const$, because we can connect this edge in fixed count combinations.

For any edge, which we connect in first step $l = const$, because we can do that in fixed count combinations, and $l'=O(n)$, because we can disconnect this edge in combinations with all edges in another circuit. Thus $max(l*l')=O(n)$.

By Theorem A2, the quantum query complexity is $\Omega\sqrt{\frac{n^2 * n^2}{n}} = \Omega(n^{1.5})$. □

Lemma L9 *If a graph $G=(V,E)$, $|V|=2n$, contains only Hamiltonian circuit then* ***Chromatic number*** *problem for two colors is solvable.*

Proof: We color one vertex in first color, next vertex in Hamiltonian circuit in second color, next in first color, etc. This coloring solve **Chromatic number** problem for two colors. □

Lemma L10 *If a graph $G=(V,E)$, $|V|=2n+1$, contains only Hamiltonian circuit then* ***Chromatic number*** *problem for two colors is not solvable.*

Proof: Regardless of how we color vertices, at least two vertices in one color will always be connected. □

Theorem T10 ***Chromatic number*** *problem requires* $\Omega(n^{1.5})$ quantum queries.

Proof: We construct the sets A and B for the usage of Theorem A1.

The set A consists of all graphs $G=(V,E)$ satisfying the requirements of Lemma L9. The set B consists of all graphs G'. Each graph G' is constructed to merge two graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$, which both satisfy the requirements of Lemma L10 and $|V_1|+|V_2|=|V|$ and $O(|V_1|)=O(|V|)$ and $O(|V_2|)= O(|V|)$.

By Lemma L8, the quantum query complexity is $\Omega(n^{1.5})$. □

4.6. Monochromatic triangle problem

We construct graphs $G=(V,E)$, $|V|=3n$ as follows. All vertices in graph are partitioned into three. Each three is connected by another three in one of the following ways (Figure 3.). Each three in this graph does not contain triangle. We put in one subset all vertices from one three.

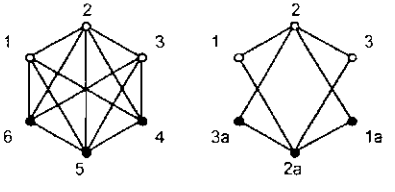


Figure3. Connections of threes

Lemma L11 *If **Monochromatic triangle** problem is not solvable in this distribution into threes, then problem is not solvable in another distribution into threes.*
Proof: We cannot mix vertices 1,2,3 with 4,5,6. If we do it three will contain triangle and **Monochromatic triangle** problem will not be solvable. If we mix vertices 1,2,3 with 1a, 2a, 3a then **Monochromatic triangle** problem solution will not change. □

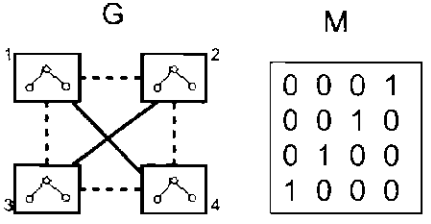


Figure 4. Graph G and matrix M

We make matrix M in accordance with graph G, which is defined before as follows. Matrix consists of n columns and n rows. We put in m_{ij} 1 if three v_i is connected with three v_j in first way (Figure 3.) and 0 if connected in second way (Figure 4.).

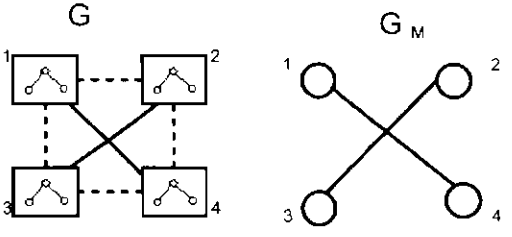


Figure5. Graph G and graph G_M

We make graph G_M, which corresponds to matrix M. Each vertex of graph G_M is related to the three of graph G (Figure 5.).

Lemma L12 *If in graph G_M, **Chromatic number** problem (K=2) is solvable, then **Monochromatic triangle** problem is solvable in corresponding graph G also.*
Proof: Each vertex of graph G_M is related to the three of graph G. Since **Chromatic number** problem is solvable, we can color G_M vertices in two colors so that one-color vertices are not connected. We put in one subset those threes in graph G, which correspond to the vertex in graph G_M colored in one color. Each three in this subset is connected with another in second way (Figure 3.) and subset does not contain triangle. □

Lemma L13 *If in graph G_M, **Chromatic number** problem (K=2) is not solvable, then **Monochromatic triangle** problem is not solvable in corresponding graph G also.*

Proof: Let us assume, that in graph G **Monochromatic triangle** problem is solvable if threes are combined differently. By Lemma L11 the problem is solvable also in this distribution of threes, which corresponds to graph G_M. But then we can solve **Chromatic number** problem in graph G_M. This is in conflict with given. □

Theorem T11 ***Monochromatic triangle*** *problem requires* $\Omega(n^{1.5})$ *quantum queries.*
Proof: By Lemma L12 and Lemma L13, complexity of **Monochromatic triangle** problem is equivalent with complexity of **Chromatic number** problem. So quantum query complexity is $\Omega(n^{1.5})$. □

References

[1] A.Ambainis. Quantum lower bounds by quantum arguments. Journal of Computer and System Sciences, 64:750-767, 2002.

[2] A.Ambainis. Personal communication. 2003.

[3] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, Umesh V. Vazirani. Strengths and Weaknesses of Quantum Computing. SIAM Journal on Computing, v. 26, 1997, pp. 1510 –1523.

[4] H. Buhrman and R. de Wolf. Complexity Measures and Decision Tree Complexity : A Survey. Theoretical Computer Science, v. 288(1): 21-43 (2002)

[5] Rūsiņš Freivalds, Andreas Winter: Quantum Finite State Transducers. SOFSEM 2001: 233-242

[6] L. Grover. A fast quantum mechanical algorithm for database search. Proceedings of the 28th ACM symposium on Theory of Computing, pp 212-219, 1996.

[7] J. Gruska. Quantum Computing. McGraw-Hill, 1999.

[8] M. Nielsen, I. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004. VOLUME 673, Datbases and Information Systems
57-65 P.

On quantum query complexity of Kushilevitz function

Aija Bērziņa and Rūsiņš Freivalds*

Institute of Mathematics and Computer Science
University of Latvia
Raiņa bulvāris 29, Riga, LV-1459, Latvia

Abstract. The query algorithms are a very convenient model for quantum complexity studies. In this paper, we show that the Kushilevitz function of 6 variables can be computed by quantum query algorithm by only 5 queries if it is allowed in some cases give wrong answer.
In deterministic case all 6 variables must be queried.

Keywords. quantum computation, query algorithms, Kushilevitz.

1. Introduction

Let $f:\{0,1\}^N \to \{0,1\}$ be a Boolean function. A *decision tree* is an algorithm for computing $f(x_1,..,x_N)$ that accesses $x_1,...,x_N$ by asking questions about the values of x_i. *The complexity* of a decision tree is the maximum number of questions that it asks. The decision tree complexity of a function f is the minimum complexity of a decision tree correctly computing f.

The theory of computation studies various models of computation: *deterministic, non-deterministic, and probabilistic and quantum* (see Papadimitriou (1994) on traditional models of computation and Gruska (1999), Nielsen and Chuang (2000) or Pittenger(1999) on quantum computation). Similarly, there are decision trees of all those types (Buhrman and de Wolf, 2001).

The decision tree complexity of f is related to *representing Boolean functions by polynomials*. Let $D(f)$ be the deterministic decision tree complexity and $Q_E(f)$ be the exact quantum decision tree complexity (see Buhrman and de Wolf, 2001, for

E-mail addresses: Aija.Berzina@tietoenator.com, Rusins.Freivalds@mii.lu.lv telephone: +371-7224363, fax: +371-820153.
Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP)

definitions). For any Boolean function f, there is a unique multilinear polynomial p such that $p(x_1, ..., x_n)=f(x_1, ..., x_n)$. $deg(f)$, the *exact* degree of a function f is the degree of corresponding polynomial p. We have $D(f) \geq deg(f)$ and $Q_E(f) \geq \frac{\deg(f)}{2}$ (The complexity of other types of decision trees is similarly related to other notions of polynomial degree which are not considered in this paper.)

Thus, if we prove that $deg(f)$ is high, this immediately implies that f is hard to compute. This observation has been used to prove that particular functions are hard to compute for both decision trees (Nisan and Szegedy, 1994; Beals et.al., 2001; Aaronson, 2002), and other models of computation (Nisan and Wigderson, 1995; Razborov, 2003).

In this paper we use traditional quantum query model. On every input x algorithm can give 2 different answers:

- 1 – if $f(x)=1$
- 0 – if $f(x)=0$

The algorithm is sometimes allowed to give wrong answer (1 instead of 0 or vice versa) but the total probability of getting right answer on every input must be > ½.

We give such an algorithm computing Kushilevitz function with 5 queries.

2. Notation and definitions

$[N]$ denotes the set $\{1, ..., N\}$.

For any boolean function $f(x_1, ..., x_N)$, there is a unique multilinear polynomial $p(x_1, ..., x_N)$ such that $f(x_1, ..., x_N)=p(x_1, ..., x_N)$ for all $x_1, ..., x_N \in \{0, 1\}$. For example, the function $f(x_1, x_2)=x_1 \; OR \; x_2$ is equal to the polynomial $x_1+x_2-x_1x_2$. $deg(f)$, the exact degree of f is just the degree of the corresponding polynomial p.

Let $D(f)$ be the deterministic decision tree complexity of f (Buhrman, de Wolf (2001)). We are interested in Boolean functions f with $deg(f)<D(f)$. Since $D(f)$ can be hard to compute, we define another quantity. $s_x(f)$, the *sensitivity* of f on input $x=(x_1, ..., x_N)$ is the number of $i \in [1, ..., N]$ such that changing the variable x_i in $(x_1, ..., x_N)$ changes the value of

$f: [f(x_1, ..., x_i, ..., x_N) \neq f(x_1, ..., 1-x_i, ..., x_N)]$

$s(f)$ is the maximum of $s_x(f)$ over all x.

CLAIM 1 (Buhrman and de Wolf, 2001) $s(f) \leq D(f)$.

In particular, if $s(f)$ is equal to the number of variables N, then $D(f)=N$. (Claim 1 implies $D(f) \geq N$. On the other hand, any f can be computed by an algorithm that queries all variables x_i.)

Fourier transform.

Let $\omega = e^{\frac{2\pi}{N}i}$ be primitive N^{th} root of unity. The Fourier transform over Z_N is given by the linear transformation F where $F_{j,k} = \frac{1}{\sqrt{N}}\omega^{jk}$

3. Kushilevitz function

Function 1 (Kushilevitz, quoted in Nisan and Wigderson (1995)).
The function $f(x_1, ..., x_6)$ of 6 variables is defined by

- $f = 0$ if the number of $x_i = 1$ is 0, 4 or 5,
- $f = 1$ if the number of $x_i = 1$ is 1, 2 or 6
- if the number of $x_i = 1$ is 3, $f = 0$ in the following cases:
  - $x_1=x_2=x_3=1$,
  - $x_2=x_3=x_4=1$,
  - $x_3=x_4=x_5=1$,
  - $x_4=x_5=x_1=1$,
  - $x_5=x_1=x_2=1$,
  - $x_1=x_3=x_6=1$,
  - $x_1=x_4=x_6=1$,
  - $x_2=x_4=x_6=1$,
  - $x_2=x_5=x_6=1$,
  - $x_3=x_5=x_6=1$

  Otherwise $f = 1$.

$deg(f)=3$, $D(f)=6$ (because the sensitivity of f on $x_1=...=x_6=0$ is 6).

We rewrite this definition in following way:

0	0	0	0	0	0	$\Rightarrow$	**0**
1	1	1	1	1	1	$\Rightarrow$	**1**
1	1	1	–	–	–	$\Rightarrow$	**0**
–	1	1	1	–	–		
–	–	1	1	1	–		
1	–	–	1	1	–		
1	1	–	–	1	–		
1	–	1	–	–	1		
–	1	–	1	–	1		

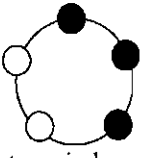
–	–	1	–	1	1
1	–	–	1	–	1
–	1	–	–	1	1

where "–" means any of symbols {0, 1}, but at least one of "–" in every row must be 0.

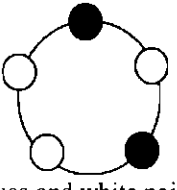
0	0	0	–	–	–		
–	0	0	0	–	–		
–	–	0	0	0	–		
0	–	–	0	0	–		
0	0	–	–	0	–		
0	–	0	–	–	0	⇒	1
–	0	–	0	–	0		
–	–	0	–	0	0		
0	–	–	0	–	0		
–	0	–	–	0	0		

where "–" means any of symbols {0, 1}, but at least one of "–" in every row must be 1.

Now it is easy to see that the "$f = 0$" block is symmetrical to the "$f = 1$" block. Moreover, if we divide a block in two parts and omit the 6^{th} column we obtain two sub-blocks where every row can be derived form another row in the sub-block by shifting it by one or more positions. So, the first sub-block corresponds to a circle



and the second sub-block corresponds to a circle



where black points denote fixed values and white points denote "–".

4. Algorithm

First, we apply a Fourier transform to a starting state to chose with equal amplitudes one of the 5 computational paths. Depending on the path chosen the first two queries are made:

1. x_1x_2
2. x_2x_3
3. x_3x_4

4. x_4x_5
5. x_5x_1

Without loss of generality we can assume that the path x_1x_2 was chosen. (Because of the symmetry all the other paths are processed in a quite similar way).
There are two possibilities:

- $x_1 = x_2$
- $x_1 \neq x_2$

1. $x_1 = x_2$

Because of symmetry we can assume that $x_1 = x_2 = 0$. Let us select corresponding rows from the blocks given above.

0	0	0	0	0	0	$\Rightarrow$	**0**
0	0	1	1	1	–	$\Rightarrow$	**0**
0	0	1	–	1	1		
0	0	0	–	–	–	$\Rightarrow$	**1**
0	0	0	0	–	–		
0	0	0	0	0	–		
0	0	–	0	0	–		
0	0	–	–	0	–		
0	0	0	–	–	0		
0	0	–	0	–	0		
0	0	0	–	0	0		
0	0	–	0	–	0		
0	0	–	–	0	0		

Now we chose with amplitude $\frac{1}{\sqrt{2}}$ one of the variables that do not have "–" in the block containing symbols "1". There are two such variables: x_3 and x_5.
Let us suppose x_3 was chosen.

1.1. $x_1 = x_2 = 0, x_3 = 1$

0	0	1	1	1	–	$\Rightarrow$	**0**
0	0	1	–	1	1		
0	0	1	0	0	–	$\Rightarrow$	**1**
0	0	1	–	0	–		
0	0	1	0	–	0		
0	0	1	0	–	0		
0	0	1	–	0	0		

Now, again with amplitude $\frac{1}{\sqrt{2}}$, we choose (x_4, x_5) or (x_6, x_5) to be queried. The pairs of values give us following answers:

- $00 \Rightarrow \mathbf{1}$
- $10 \Rightarrow \mathbf{1}$
- $11 \Rightarrow \mathbf{0}$
- $01 \Rightarrow \mathbf{0}$ with amplitude $\frac{1}{\sqrt{2}}$, and **1** with amplitude $\frac{1}{\sqrt{2}}$

1.2. $x_1 = x_2 = 0, x_3 = 0$

0	0	0	0	0	0	$\Rightarrow$	**0**
0	0	0	–	–	–		
0	0	0	0	–	–		
0	0	0	0	0	–		
0	0	0	0	0	–		
0	0	0	–	0	–		
0	0	0	–	–	0	$\Rightarrow$	**1**
0	0	0	0	–	0		
0	0	0	–	0	0		
0	0	0	0	–	0		
0	0	0	–	0	0		

We choose (x_4, x_6) or (x_5, x_6) to be queried with amplitude $\frac{1}{\sqrt{2}}$. The pairs of values give us following answers:

- $00 \Rightarrow \mathbf{0}$ with amplitude $\frac{1}{\sqrt{2}} + \varepsilon$, and **1** with amplitude $\frac{1}{\sqrt{2}} - \varepsilon$, $\varepsilon > 0$.
- $01 \Rightarrow \mathbf{1}$
- $10 \Rightarrow \mathbf{1}$
- $11 \Rightarrow \mathbf{1}$

2. $x_1 \neq x_2$

Without loss of generality we can assume that $x_1 = 0$ and $x_2 = 1$. And again we select corresponding rows from the blocks given above.

0	1	1	1	–	–		
0	1	1	1	1	–		
0	1	–	1	–	1	$\Rightarrow$	**0**
0	1	1	–	1	1		
0	1	–	–	1	1		
0	1	0	0	0	–	$\Rightarrow$	**1**
0	1	–	0	0	–		

0	1	0	–	–	0
0	1	0	–	0	0
0	1	–	0	–	0

In this case we always ask the value of x_4.
There can be two cases:

2.1. $x_1 = 0, x_2 = 1, x_4 = 1$

0	1	1	1	–	–		
0	1	1	1	1	–		
0	1	–	1	–	1	⇒	**0**
0	1	1	1	1	1		
0	1	–	1	1	1		
0	1	0	1	–	0	⇒	**1**
0	1	0	1	0	0		

Next, values of (x_3,x_6) must be queried. The pairs of values give us following answers:

- $00 \Rightarrow 1$
- $01 \Rightarrow 0$
- $10 \Rightarrow 0$
- $11 \Rightarrow 0$

2.2. $x_1 = 0, x_2 = 1, x_4 = 0$

0	1	1	0	1	1	⇒	**0**
0	1	–	0	1	1		
0	1	0	0	0	–		
0	1	–	0	0	–		
0	1	0	0	–	0	⇒	**1**
0	1	0	0	0	0		
0	1	–	0	–	0		

We ask values of (x_5, x_6). The pairs of values give us following answers:

- $00 \Rightarrow 1$
- $01 \Rightarrow 1$
- $10 \Rightarrow 1$
- $11 \Rightarrow 0$

Amplitude of every answer is $\frac{1}{\sqrt{5}}$

Let us now count the probabilities of getting the right answer on different inputs. It is important to know on how many computational paths every input will be processed like in $x_1 = x_2$ case and on how many paths it will be processed like $x_1 \neq x_2$ case.

All the inputs can be divided into following parts by the first variables:

- All the first 5 variables are equal. This corresponds to 4 input strings:

0	0	0	0	0	0
1	1	1	1	1	1
0	0	0	0	0	1
1	1	1	1	1	0

These inputs every time will be processed like in $x_1 = x_2$ case. For 000000 and 111111 we will every time get the answer **0** with amplitude $\frac{1}{\sqrt{20}}\left(\frac{1}{\sqrt{2}}+\varepsilon\right)$, and **1** with amplitude $\frac{1}{\sqrt{20}}\left(\frac{1}{\sqrt{2}}-\varepsilon\right)$. This gives us the total probability of answer **0** = $\left(\frac{1}{2}+\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)>\frac{1}{2}$ and the total probability of answer **1** = $\left(\frac{1}{2}-\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)>\frac{1}{2}$.

For 000001 and 111110 we get the right answer with total probability = 1.

- One of the first 5 variables differs from others. There are 20 such strings. Each of them is 3 times processed like $x_1 = x_2$ case and 2 times like $x_1 \neq x_2$ case.

In $x_1 = x_2$ case we get the right answer two times with probability $\frac{3}{20}+\frac{1}{20}\left(\frac{1}{2}-\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)$ and once with probability $\frac{2}{20}+\frac{2}{20}\left(\frac{1}{2}-\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)$. This gives us $\frac{5}{20}+\frac{3}{20}\left(\frac{1}{2}-\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)$. In $x_1 \neq x_2$ case we every time get the right answer with amplitude $\frac{1}{\sqrt{5}}$. So the probability definite answer in this case is $\frac{13}{20}+\frac{3}{20}\left(\frac{1}{2}-\frac{1}{\sqrt{2}}\varepsilon+\varepsilon^2\right)>\frac{1}{2}$.

- Two of the first 5 variables are equal and there is no variable between them. (For example, 001100) There are 20 such strings. Each of them is 3 times processed like $x_1 = x_2$ case and 2 times like $x_1 \neq x_2$ case.

In $x_1 = x_2$ case we once get the right answer with probability $\frac{3}{20}$. All the other cases give $\frac{16}{20}$. so the total probability is $\frac{19}{20}$.

- Two of the first 5 variables are equal and there is a variable between them. (For example. 001010) There are 20 such strings. Each of them is once processed like $x_1 = x_2$ case and 4 times like $x_1 \neq x_2$ case. In $x_1 = x_2$ case we get the right

answer with probability $\frac{1}{10}$, the other cases we give $\frac{4}{5}$, so the total probability of right answer is $\frac{9}{10}$.

References

[1] Aaronson, S. (2002); Quantum lower bound for the collision problem. *Proceedings of the 32nd ACM Symposium on Theory of Computing*, ACM Press, Montreal, pp. 635--642.

[2] Ambainis, A. (2003); Polynomial degree vs. quantum query complexity. *Proceedings of the 44th IEEE Conference on Foundations of Computer Science*,IEEE Press, Boston.

[3] Ambainis, A. (2003a); Quantum query algorithms and lower bounds. *Proceedings of the 3rd EuroConference on Foundations of Formal Sciences*, Kluwer Academic Publishers. Vienna.

[4] Beals, R., Buhrman, H., Cleve, R., Mosca, M., de Wolf, R. (2001) Quantum lower bounds by polynomials. *Journal of ACM, **48**, pp. 778--797.*

[5] Buhrman, H., de Wolf, R. (2001) Complexity measures and decision tree complexity: a survey.

Theoretical Computer Science, **288**, pp. 21--43.

[6] Coulborn, C. J., Dinitz, J. H. (1996) *The CRC Handbook of Combinatorial Designs*,CRC Press, Boca Raton, 784 pp.

[7] Gruska, J. (1999) *Quantum Computing. McGraw-Hill, London, 439 pp.*

[8] Hall, M. (1998) *Combinatorial Theory. 2nd edition*,J. Wiley, New York, 464 pp.

[9] Nielsen, M., Chuang, I. (2000) *Quantum Computation and Quantum Information*.Cambridge University Press, New York, 700 pp.

[10] Nisan, N., Szegedy, M. (1994) On the degree of Boolean functions as real polynomials. *Computational Complexity*, **4**, pp. 301--313.

[11] Nisan, N., Wigderson, A. (1995) On rank vs. communication complexity. *Combinatorica*, **15**, pp. 557--565.

[12] Papadimitriou, C. (1994) *Computational Complexity*. Addison-Wesley, Reading, 500 pp.

[13] Pittenger, O. (1999) *An Introduction to Quantum Computing Algorithms*.Birkhauser. Boston, 152 pp.

[14] Razborov, A. (2003) Quantum communication complexity of symmetric predicates, *Izvestiya of the Russian Academy of Science, Mathematics*, **67**, pp. 159--176.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
66–72 P.

Quantum query algorithm complexity for graph circuit problem

Vasilijs Kravcevs

University of Latvia
md80004@lanet.lv

Abstract. In this paper we observe graph circuit problem. We present quantum query algorithm that solves this problem with query complexity $O(n^{1.5})$. We also prove lower bound of complexity for quantum query algorithms that solve this problem. We use Ambainis' lower bound technique to prove the results of the work.

1. Introduction

The speedups of quantum algorithms over classical algorithms have been a main reason for the current interests on quantum computing. The central question of the quantum computing is: how powerful the quantum algorithms are, how much speedup is possible? Query model is often used in studying quantum complexity, because many known quantum algorithms fall into this framework. The query (or black-box, or oracle) model of computation arises when one is given a block-box containing an N-tuple of Boolean variables $X = (x_1, x_2, \ldots, x_N)$. The box is equipped to output x_i on input i. The input is accessed, only by querying an oracle, and the goal is to minimize the number of queries made. Usually we are interested in bound-error computation, where the output is correct with probability at least $2/3$ for all inputs. We use $Q_2(f)$ to denote minimal number of queries for computing f with bound-error.

Two main lower bound techniques for $Q_2(f)$ are the polynomial method [3] and adversary method [1]. We use a result by A. Ambainis [1,2] and S. Zhang [7] to prove lower complexity bounds for quantum query algorithm that solves graph circuit problem.

2. Definitions

In the query model, the input $x_1, x_2, \ldots, x_N$ is contained in a black-box and can be accessed by queries to the black-box. The classical version of the query model is known as *decision trees* [5]. A *quantum decision tree* with T queries is a quantum black-box model, where queries and other operations can be made in quantum superposition. Such decision tree can be represented as sequence of unitary transformations:

$$U_0, O, U_1, O, \ldots. U_{T-1}, O, U_T$$

where the U_i are arbitrary unitary transformations, and the O are unitary transformations which correspond to the queries to oracle. The computation starts

with state $|0\rangle$. Then we apply $U_0,O,U_1,O, \ldots,U_{T-1},O,U_T$. The computation ends with some measurement of the final state. The measurement gives us the probability of receiving the value of the function f(X).

The quantum algorithm computes a function $f(x_1, x_2, \ldots, x_N)$ with bound-error if for every $x = (x_1, x_2, \ldots, x_N)$ for which f is defined, the probability that the rightmost bit of $U_TOU_{T-1}O\ldots U_1OU_0|0\rangle$ equals $f(x_1, x_2, \ldots, x_N)$ is at least $\frac{2}{3}$. The query complexity of f is the smallest number of queries used by a quantum algorithm that computes f with bound-error. We denote it $Q_2(f)$.

Our proof of lower bound uses the following results by A. Ambainis [1,2] and S. Zhang [7].

Theorem 1. *[1] Let $f: \{0,1\}^N \to \{0,1\}$ be a function and X, Y be two sets of inputs s. t. $f(x) \neq f(y)$ if $x \in X$ and $y \in Y$. Let $R \subseteq X \times Y$ be a relation s.t.*

1. *$\forall x \in X$, there are at least m different $y \in Y$ s.t. $(x,y) \in R$.*
2. *$\forall y \in Y$, there are at least m' different $x \in X$ s.t. $(x,y) \in R$.*
3. *$\forall x \in X$, $\forall i \in [N]$, there are at most l different $y \in Y$ s.t. $(x,y) \in R$ and $x_i \neq y_i$.*
4. *$\forall y \in Y$, $\forall i \in [N]$, there are at most l' different $x \in X$ s.t. $(x,y) \in R$ and $x_i \neq y_i$.*

Then $Q_2(f) = \Omega\left(\sqrt{\frac{mm'}{ll'}}\right)$.

Definition 1. *[2] Let $f: \{0,1\}^N \to \{0,1\}$ be a Boolean function and X, Y be two sets of inputs s. t. $f(x) \neq f(y)$ if $x \in X$ and $y \in Y$. Let $R \subseteq X \times Y$ be a relation. A weight scheme for X, Y, R consists of three weight functions $w(x,y) > 0$, $u(x,y,i) > 0$ and $v(x,y,i) > 0$ satisfying*

$$u(x,y,i)v(x,y,i) \geq w^2(x,y)$$

for all $(x,y) \in R$ and $i \in [N]$ with $x_i \neq y_i$. We further denote

$$w_x = \sum_{y:(x,y)\in R} w(x,y), \qquad w_y = \sum_{x:(x,y)\in R} w(x,y)$$

$$u_{x,i} = \sum_{y:(x,y)\in R, x_i \neq y_i} u(x,y,i), \qquad v_{y,i} = \sum_{x:(x,y)\in R, x_i \neq y_i} v(x,y,i).$$

Theorem 2. *[2]Let $f: \{0,1\}^N \to \{0,1\}$ be a function and $X \subseteq f^{-1}(0)$, $Y \subseteq f^{-1}(1)$ and $R \subseteq X \times Y$. Let w, u, v be a weight scheme for X, Y, R. Then*

$$Q_2(f) = \Omega\left(\sqrt{\min_{x\in X, i\in[N]} \frac{w_x}{u_{x,i}} \min_{y\in Y, i\in[N]} \frac{w_y}{v_{y,i}}}\right)$$

It is not hard to see, that Theorem 2. generalizes Theorem 1. Later S. Zhang has improved the results of A. Ambainis and proved the theorem that generalizes Theorem 2.

Theorem 3. *[7]Let f: $\{0,1\}^N \to \{0,1\}$ be a function and $X \subseteq f^{-1}(0)$, $Y \subseteq f^{-1}(1)$ and $R \subseteq X \times Y$. Let w, u, v be a weight scheme for X, Y, R. Then*

$$Q_2(f) = \Omega\left(\sqrt{\min_{(x,y)\in R, i\in[N], x_i\neq y_i} \frac{w_x w_y}{u_{x,i} v_{y,i}}}\right)$$

3. Graph circuit problem

Problem 1. Graph circuit problem.
INSTANCE: Graph G = (V, E), E|=n, represented by adjacency matrix A = (a_{ij}), where

$$a_{ij} = \begin{cases} 1, \text{ if there is edge between vertices i and j} \\ 0, \text{ otherwise} \end{cases}$$

QUESTION: Is there any circuit in this graph G?

It is easy to see that query complexity of classical algorithm is equal with $O(n^2)$ (because we need to query all values of the matrix in the worst case). We will prove the complexity of quantum algorithm that is better than in classical case.

Lemma 1. *If there are more than (n-1) edges in G = (V,E), then there is circuit in this graph.*
Proof. Easy to see that from all graphs without circuits the maximal number of edges is in the graph that equals with graph's spanning tree, but for spanning tree: |E| = |V|-1 = n-1. And if we add one more edge to the spanning tree, there will be circuit in acquired graph.

Theorem 4. *The quantum query algorithm complexity for Problem 1. is $O(n^{1.5})$.*

Proof. We construct the algorithm with quantum query complexity $O(n^{1.5})$ that solves the Problem 1.

Algorithm.

1) Fix $k = 1$, which denotes current row in adjacency matrix, that algorithm is processing. Fix $l = 1$ – the column in a row from which forward we will look for ones (edges) in that row. Fix $m = 0$, which denotes number of edges that we have found.
2) In the k row we search with Grover's search algorithm for $j > l$, such that a_{kj} equals with 1. If there is such a_{kj} then go to step 3, if there is no such a_{kj}, then increase k by 1 (we need to search in the next row). If $k > n$, then go to the step 4, otherwise: assign k to l (graph is not directed, so we do not need to search for ones in the first k columns in the next row), and repeat the second step.
3) Remember pair (k,j) and increase m by 1. If $m = n$, then return "Yes" and finish the algorithm; if $m < n$, then assign j to l and go the second step (continue searching for edges in the current row).

4) Search for circuit in the graph using classical deterministic algorithm that takes an input all the pairs (k,l) that we were remembered in previous steps. The output of our algorithm will be the same as the output of classical algorithm.

It is easy to see that algorithm is correct. Indeed, in steps 2 and 3 of the algorithm we search for graph edges, and we finish searching whether when we find n edges (step 3) or when we achieve the end of the adjacency matrix (if there are less than n edges in the graph). If we find n edges then, by Lemma 1, there is circuit in the graph, so the output is "Yes". If the number of edges is less than n, then classical algorithm with input all the edges, we have found, will output the right answer.

Query complexity of the algorithm.

It is easy to prove that number of the queries in the algorithm is $O(n\sqrt{n})$. Indeed, Grover's search in the second step of the algorithm is running less than 2n times, because for every edge, that we have found, we use one Grover's search, but the number of found edges $\leq$ n; and for every row we use one extra Grover's search, when there is no 1 anymore in the row. So the total number of Grover's searches is less than 2n. The query complexity of each Grover's search is $O(\sqrt{n})$ because the size of the input $\leq$ n. So the total query complexity of the algorithm is $O(n\sqrt{n})$.

□

We can see that algorithm we have constructed not only detect whether there is a circuit in the graph, but also find a set of edges that contains a circuit if there is any.

We use adversary lower bound technique by A. Ambainis to prove the lower bound of complexity for quantum query algorithms that solve the Problem 1.

Theorem 5. *The quantum query algorithm complexity for Problem 1 is* $\Omega(n)$.

Proof.

We construct such sets X and Y for the usage of Theorem 1. The set X consists of all spanning trees (see fig. 1.a) and f(X) = 0. The set Y consists of all such graphs that were constructed adding one extra edge to the spanning tree (see fig. 1.b). By Lemma 1, f(Y) = 1. The relation $R \subseteq X \times Y$ consists of pairs (x,y) such that y is built from x adding one extra edge to x.

From each graph $x \in X$ we can obtain graph $y \in Y$ such that $(x,y) \in R$ by adding one edge to graph x. Hence $m = C_n^2$ - (n-1) - $\frac{n(n-1)}{2}$ - (n-1) = $\frac{(n-1)(n-2)}{2}$ = $O(n^2)$ because we can put an edge between any two vertices except those edges that already exist. From each graph $y \in Y$ we can obtain graph $x \in X$ such that $(x,y) \in R$ by disconnecting any one of the edges in the circuit. Hence $m' = O(1)$. $l = l' = 1$. Hence, by Theorem 1, $Q_2(f) = \sqrt{mm'} = \Omega(\sqrt{n^2}) = \Omega(n)$.

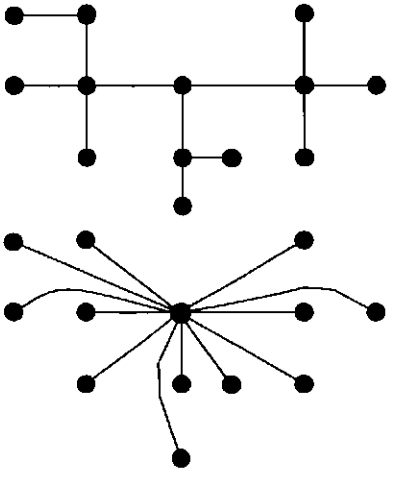
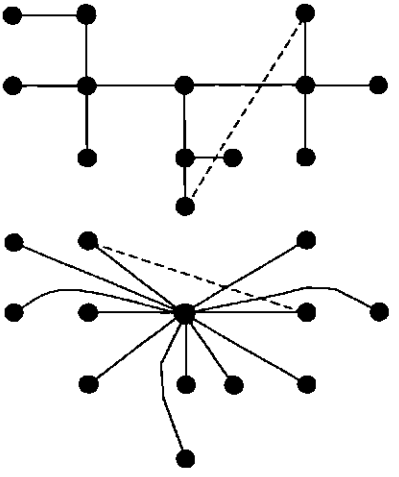


Fig.1.a) Instances of the set X



b) Instances of the set Y

As we can see, the proved lower bound is not tight. But, using the Theorem 3, we can prove that every quantum algorithm that solves graph's circuit problem needs to make $O(n^{1.5})$ queries to the oracle. So, we will show that our algorithm is optimal.

Theorem 6. *The quantum query algorithm complexity for Problem 1 is* $\Omega(n^{1.5})$.

Proof.

We construct such sets X and Y for the usage of Theorem 3. The set X consists of all such graphs that obtained from graphs of unique n-circuit by removing one edge (see fig. 2.a). The set Y is the set of all graphs that constructed from graphs with

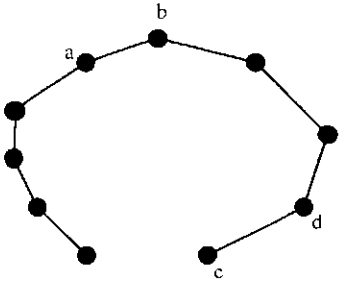
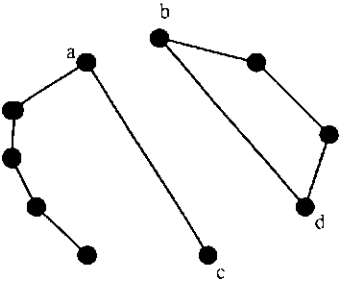


Fig. 2. a) instances of the set X



b) instances of the set Y

exactly two circuits each of length between $n/3$ and $2n/3$ by removing one egde from one of the circuits (see fig. 2.b). The relation $R \subseteq X \times Y$ consists of pairs (x,y) such that only difference between x and y is that exists four vertices a, b, c, d such that edges (a,b) and (c,d) are in x but not in y and edges (a,c) and (b,d) are in y but not in x (see fig. 2). A weight scheme for X, Y, R consists of three weight functions

w(x,y) =1, u(x,y,(i,j)) = $\sqrt{n}$ and v(x,y,(i,j)) = $1/\sqrt{n}$, for all (x,y) $\in$ R and (i,j) $\in$ $[n^2]$ with $x_{(i,j)} \neq y_{(i,j)}$ (by $x_{(i,j)}$ we denote the graph's adjacency matrix's (i,j) value). The property of weight functions is valid, because $\sqrt{n}\frac{1}{\sqrt{n}} = 1 \geq 1 = w^2(x,y)$.

$w_x = \sum_{y:(x,y)\in R} w(x,y)$ = O(n^2), because there are O(n^2) such y that (x,y)$\in$ R: (n-2) opportunities to choose edge (a,b) and then $n/3$ opportunities for the second edge (c,d).

$w_y = \sum_{x:(x,y)\in R} w(x,y)$ = O(n^2), because there are O(n^2) such x that (x,y)$\in$ R: from each part of the graph y we need to remove one edge (but length of the part is at least $n/3$).

$u_{x,(i,j)} = \sum_{y:(x,y)\in R, x_{(i,j)}\neq y_{(i,j)}} u(x,y,(i,j)) = \sum_{y:(x,y)\in R, x_{(i,j)}=0 \& y_{(i,j)}=1} u(x,y,(i,j)) + \sum_{y:(x,y)\in R, x_{(i,j)}=1 \& y_{(i,j)}=0} u(x,y,(i,j))$.

$\sum_{y:(x,y)\in R, x_{(i,j)}=0 \& y_{(i,j)}=1} u(x,y,(i,j)) = 4\sqrt{n}$, because if there is no edge between i and j in x, but such edge exists in y, that means that this is the case of vertices a, c or vertices b, d and the endpoints of the second edge must be neighbors of them, so we have just 4 opportunities to choose neighbors.

$\sum_{y:(x,y)\in R, x_{(i,j)}=1 \& y_{(i,j)}=0} u(x,y,(i,j))$ = O($n\sqrt{n}$), because edge (i,j) is the edge (a,b) (or edge (c,d)) and we have $n/3$ opportunities for removing second edge (c,d) (or edge (a,b) respectively).

Similarly:

$\sum_{x:(x,y)\in R, y_{(i,j)}=0 \& x_{(i,j)}=1} v(x,y,(i,j)) = 4/\sqrt{n}$, and

$\sum_{x:(x,y)\in R, y_{(i,j)}=1 \& x_{(i,j)}=0} v(x,y,(i,j))$ = O($n/\sqrt{n}$).

$$\sqrt{\min_{(x,y)\in R, i\in[n^2], x_{(i,j)}\neq y_{(i,j)}} \frac{w_x w_y}{u_{x,(i,j)} v_{y,(i,j)}}} =$$

$$\sqrt{\min\left(\min_{(x,y)\in R, i\in[n^2], x_{(i,j)}=0 \& y_{(i,j)}=1} \frac{w_x w_y}{u_{x,(i,j)} v_{y,(i,j)}}, \min_{(x,y)\in R, i\in[n^2], x_{(i,j)}=1 \& y_{(i,j)}=0} \frac{w_x w_y}{u_{x,(i,j)} v_{y,(i,j)}}\right)} =$$

$$\sqrt{\min\left(\frac{O(n^2)\cdot O(n^2)}{4\sqrt{n}\cdot O(n/\sqrt{n})}, \frac{O(n^2)\cdot O(n^2)}{O(n\sqrt{n})\cdot 4/\sqrt{n}}\right)} = \sqrt{\min\left(\frac{O(n^4)}{O(n)}, \frac{O(n^4)}{O(n)}\right)} =$$

$\sqrt{\min(O(n^3), O(n^3))} = O(n^{1.5})$. So, by Theorem 3. $Q_2(f) = \Omega(n^{1.5})$.

□

4. Conclusion

In this paper we have observed graph circuit problem. We have constructed quantum algorithm with query complexity $O(n^{1.5})$ and have shown that this algorithm is optimal, proving that the lower bound of complexity for quantum query algorithms that solve this problem is $\Omega(n^{1.5})$.

References

1. A. Ambainis. Quantum lower bounds by quantum arguments, *Jornal of Computer and System Sciences*, 64:750-767,2002.
2. A. Ambainis. Polynomial degree vs. quantum query complexity, *FOCS'03*, 2003.
3. R. Beals, H. Buhrman, R. Cleve, M. Mosca and R. Wolf. Quantum lower bounds by polynomials. *Journal of ACM*, 48: 778-797, 2001.
4. A. Berzina, A. Dubrovsky, L. Lace, R. Freivalds, O. Scegulnaja. Quantum query complexity for some graph problems, *SOFSEM 2004*: 140-150.
5. H. Buhrman and R. Wolf. Complexity Measures and Decision Tree Complexity: A Survey. *Theoretical Computer Science*, v. 288(1): 21-43, 2002.
6. L. K. Grover. A fast quantum mechanical algorithm for database search. *In Proceedings of 28th STOC*, pages 212-219, 1996.
7. S. Zhang. On the power of Ambainis's lower bounds, *quant-ph/0311060*.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
73–80 P.

Better Probabilities for Quantum One-Way Finite Automata with Counter

Maksim Kravtsev[1]

Department of Computer Science,
University of Latvia, Raina bulv. 19, Riga, Latvia**.
maksims@batsolf.lv

Abstract. The paper considers recognition of several non-context-free languages with one-counter one-way finite automaton. These languages were introduced in [7]. The results of the paper is further improvement of probabilities shown for these languages in [8].

1 Introduction

Quantum automata were introduced by Kondacs and Watrous [6], who showed that 2-way quantum automata recognize all regular languages and some non-regular languages such as 0^n10^n, and are thus more powerful than their 2-way probabilistic counterparts. The same authors showed, however, that the opposite relation holds for the quantum one-way finite automata, which can recognize only the proper subset of regular languages. It can not recognize for example the language $\{0,1\}^*1$. This restriction comes from the requirements for reversibility.

There are considered several models in classical computing supplied with some additional enhancements like multitape and pushdown automata; one of the simplest such models is one-way automaton supplied with a counter. The quantum version of automata with counter (1Q1CA) was considered by Kravtsev [7]. Further study of such automata are given in [8] where probabilistic reversible version automata are also considered: [3] where the language recognizable with 1Q1CA and not recognizable with probabilistic are shown: [9] where properties of two-way automata with counter are considered.

** Research supported by the Latvian Council of Science

In this paper we consider several non-context-free languages from [7]. Probabilities of their recognition were improved in [8] using probabilistic approach. In this paper we combine techniques of [7] and [8] to further improve probabilities of their recognition.

2 Definitions

Automata. See Gruska [5] for general and Condacs and Watrous [6] for technical background material on quantum finite automata. The quantum automata with counter which we consider here, were considered by Kravtsev [7], which we also refer to for formal definitions. Here, we recall the general ideas of the model.

To begin with, a *deterministic* one-way finite automaton with counter (1D1CA) is specified by a finite *input alphabet* Σ, a finite set Q of *states* with a singled out *initial state*, two disjoint sets of *accepting* and *rejecting* states Q_a and Q_r, a *counter* which is allowed to hold an arbitrary integer, and a *transition function* δ updating the state and the counter at each step of computation as input letters are read from the tape. The value of the function is dependent of the current state, the read letter and whether the counter is zero or non-zero, but not the exact value of the counter. The counter is set to 0 at the beginning of computation; it is allowed to change by one at each update at most. The automaton processes each letter of the input word precisely once until the last letter of the word is reached. If the automaton is then in an accepting state, the word is considered accepted, while if in a rejecting state, the word is rejected.

Formally δ is defined as mapping $Q \times \Sigma \times S \to Q \times D$, where $S = \{0, 1\}$ denotes whether the counter is 0 or not 0 and $D = \{-1, 0, 1\}$ denotes the number by which the value of the counter changes.

A *probabilistic* version of this type of automaton (1P1CA) is obtained in the usual way, essentially by considering the states of the deterministic case as point masses in a larger set of probability measures, to which the evolution of the automaton is then extended. Formally we define δ in this case as $\delta: Q \times \Sigma \times S \times Q \times D \to R^+$, where $\delta(q, \sigma, s, q', d)$ describes probability of getting from the state q and the value of the counter is 0 or not 0 s by reading σ letter, to

state q' and change the value of the counter by d. δ should satisfy the following condition: $\sum_{q',d} \delta(q, \sigma, s, q', d) = 1$ for each $q, q' \in Q$, $\sigma \in \Sigma$, $s \in \{0, 1\}$, $d \in \{-1, 0, 1\}$

Finally, for the one-way *quantum* one-counter automaton (1Q1CA), the alphabet is supplemented by end markers $\not\rightarrow$ and $\not\leftarrow$. transition function δ defined as $\delta: Q \times \Gamma \times S \times Q \times D \rightarrow C$, is assumed to satisfy certain well-formedness conditions which ensure that the evolution of the automata is a unitary transformation. Here $\Gamma = \Sigma \cup \{\not\rightarrow, \not\leftarrow\}$. The definition of the counter and the actions on it remain unchanged from the classical version.

Language recognition works roughly as follows: for each letter σ of a word extended by the end markers, a unitary operator U_σ is applied to the current state (configuration) of automaton and the resulting state (configuration) is observed using an observable $l_2(C_n) = E_a, \oplus E_r \oplus E_$of rejecting, accepting, and non-terminating subspaces. The state then collapses into one of the subspaces $E_a, E_r, E\_$. If a "non-terminating" state is observed, the computation continues with the next letter. The probability of the acceptance, rejection and non-termination at each step is equal to the square of the norm of the corresponding collapsed state.

Quantum state. More formally the state of 1Q1CA is considered as normalized vector of Hilbert space $l_2(Q \times \mathbb{Z})$ with basis vectors $|q, k\rangle$ corresponding to the possible configurations of the classical automaton - the combinations of state $q \in Q$ and counter value $k \in \mathbb{Z}$. So state of quantum automaton is a superposition $\sum_{ik} a_{ik} |q_i, k\rangle$ where $a_{ik} \in C$ is complex amplitude of vector $|q, k\rangle$ and $\sum_{ik} a_{ik}^2 = 1$.

Transformation. Operator U_σ in terms of δ on the vector $|q, k\rangle$ acts as follows

$$U_\sigma^\delta |q, k\rangle = \sum_{q',d} \delta(q, \sigma, sign(k), q', d) |q', k + d\rangle ,$$

where $\mathrm{sign}(k) = 0$ if $k = 0$ and 1 otherwise. By linearity U_σ^δ can be applied to any superposition of basis states.

To ensure that U_σ^δ is unitary δ should satisfy the following conditions of well-formedness [7]. For each $q_1, q_2, q' \in Q, \sigma \in \Gamma, s \in \{0, 1\}, d \in \{-1, 0, 1\}$:

$$\sum_{q',d} \delta^*(q_1, \sigma, s_1, q', d) \delta(q_2, \sigma, s_2, q', d) = \begin{cases} 1, & if\ q_1 = q_2 \\ 0, & if\ q_1 \neq q_2 \end{cases}$$

$$\sum_{q',d} \delta^*(q_1,\sigma,s_1,q',1)\delta(q_2,\sigma,s_2,q',0)+$$

$$+\sum_{q',d} \delta^*(q_1,\sigma,s_1,q',0)\delta(q_2,\sigma,s_2,q',-1) = 0$$

$$\sum_{q',d} \delta^*(q_1,\sigma,s_1,q',1)\delta(q_2,\sigma,s_2,q',-1) = 0,$$

where * denotes complex conjunctive.

Observable. Mathematically the result of observation of superposition $\sum_{ik} a_{ik}|q_i,k\rangle$ is the following: probability to accept is $p_a = \sum_{ik} a_{ik}^2$ where $q_i \in Q_a$, probability to reject is $p_r = \sum_{ik} a_{ik}^2$ where $q_i \in Q_r$ and computation halts in these cases, otherwise the computation continues with the following state $\sum_{ik} \frac{a_{ik}}{\sum_{jk} a_{jk}}|q_i,k\rangle$ where $q_i, q_j \in Q_-$

Simple one-way one-counter quantum automaton. 1Q1CA is *simple* if its transition function is defined as follows:
1) the set of finite unitary matrices, that map states from Q to states from Q, different for each input letter and zero or not zero counter value.
2) the change of the value of the counter is determined by the read letter and the state from Q, that automaton moves in.

A QF1CA is *simple*, if for each $\sigma \in \Gamma, s \in \{0,1\}$ there is a linear unitary operator $V_{\sigma,s}$ on the inner product space $l_2(Q)$ and a function $D: Q, \Gamma \to \{-1,0,1\}$ such as for each $q, q' \in Q$

$$\delta(q,\sigma,s,q',d) = \begin{cases} \langle q'|V_{\sigma,s}|q\rangle & if\ D(q',\sigma) = d \\ 0 & else \end{cases}$$

where $\langle q'|V_{\sigma,s}|q\rangle$ denotes the coefficient of $|q'\rangle$ in $V_{\sigma,s}|q\rangle$.

As shown in [7] the *simple* 1Q1CA satisfies the well-formedness conditions. Further in the paper we work with simple automata.

3 Results

Languages. Consider the alphabet $\Sigma = \{0,1\}$.

The language L_1. $L_1 = \{\ 0^i10^j10^k \parallel$ (i=k or j=k) and $\neg$(i=j)) $\}$.

The language L_2. $L_2 = \{\ 0^i10^j10^k \parallel$ exactly 2 of i,j,k are equal$\}$.

Theorem 1. *The language* L_1 *can be recognized by quantum one-counter one-way automata with probability* $\frac{5}{8}$.

Proof. Let $V_{\hookrightarrow,0}|q_0\rangle = \frac{\sqrt{5}}{4}|q_{0,i=k}\rangle + \frac{\sqrt{5}}{4}|q_{0,j=k}\rangle + \frac{\sqrt{6}}{4}|q_{0,k=(i+j)/2}\rangle$, where q_0 is initial state, $q_{0,j=k}$, $q_{0,i=k}$ and $q_{0,k=(i+j)/2}$ non-terminating states. Transitions for 0 and 1 can be defined in such way, that it would be reversible and deterministic and the following conditions are satisfied if starting state of such deterministic automaton is $q_{0,j=k}$, $q_{0,i=k}$ or $q_{0,k=(i+j)/2}$ respectively:
1) If the word is of form $0^i10^j10^k$ that no rejection or acceptance occur during the computation and the $q_{0,j=k}$ leads to the state $q_{j=k}$ and counter equal to the j-k, the $q_{0,i=k}$ to $q_{i=k}$ and counter i-k, the $q_{0,k=(i+j)/2}$ to $q_{k=(i+j)/2}$ and counter k - (i+j)/2.
2) If the word is not like $0^i10^j10^k$ than the word is rejected in each path, or is in some other state then $q_{i=k}$, $q_{j=k}$ and $q_{k=(i+j)/2}$ and thus will be rejected on end marker $\hookleftarrow$.
Such transitions for example for $q_{0,j=k}$, will be as follows: 4 states $q_{0,j=k}$, $q_{1,j=k}$, $q_{2,j=k}$, $q_{3,j=k}$. Transitions for 0 are defined as each state remains the same and counter value is increased if resulting state is $q_{1,j=k}$, and decreased if $q_{2,j=k}$: transitions for 1 defined as states should shift to the next index in respect with module 4 so $q_{n,j=k}$ should shift to $q_{n+1 mod 4,j=k}$ and counter value remains the same. $q_{3,j=k}$ should be rejecting state, $q_{2,j=k} = q_{j=k}$.
We define V for $\hookleftarrow$ as
$V_{\hookleftarrow,0}\ |q_{i=k}\rangle = \sqrt{\frac{3}{5}}|q_{a1}\rangle + \frac{1}{\sqrt{5}}|q_a\rangle + \frac{1}{\sqrt{5}}|q_r\rangle$;
$V_{\hookleftarrow,0}\ |q_{j=k}\rangle = \sqrt{\frac{3}{5}}|q_{a2}\rangle - \frac{1}{\sqrt{5}}|q_a\rangle + \frac{1}{\sqrt{5}}|q_r\rangle$;
$V_{\hookleftarrow,0}\ |q_{k=(i+j)/2}\rangle = |q_{r2}\rangle$;
$V_{\hookleftarrow,1}\ |q_{k=(i+j)/2}\rangle = |q_{a2}\rangle$
where q_a, q_{a1}, q_{a2} are accepting states and q_r , q_{r2} are rejecting states. All the other transition should be defined to map any of non-halting states to the rejecting states (it can be done by adding some more rejecting states to retain unitarity see [7]).
Let us consider how the computation goes with such automata. Before reading $\hookleftarrow$ word is rejected if it is not of kind $0^i10^j10^k$, and

is in the superposition
$|q'\rangle = \frac{\sqrt{5}}{4}|q_{i=k}, i-k\rangle + \frac{\sqrt{5}}{4}|q_{j=k}, j-k\rangle + \frac{\sqrt{6}}{4}|q_{k=(i+j)/2}, k-(i+j)/2\rangle$
otherwise. We should consider following cases then:

- 1. If i=j=k then the state after reading $\vdash\rho$ becomes
  $\frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}}|q_{a1},0\rangle + \frac{1}{\sqrt{5}}|q_a,0\rangle + \frac{1}{\sqrt{5}}|q_r,0\rangle\right) +$
  $+ \frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}}|q_{a2},0\rangle + \frac{1}{\sqrt{5}}|q_a,0\rangle + \frac{1}{\sqrt{5}}|q_r,0\rangle\right) + \frac{\sqrt{6}}{4}|q_r,0\rangle =$
  $\frac{\sqrt{3}}{4}|q_{a1},0\rangle + \frac{\sqrt{3}}{4}|q_{a2},0\rangle + \frac{1}{2}|q_r,0\rangle + \frac{\sqrt{6}}{4}|q_r,0\rangle.$
  Thus the total probability of rejection, by summing squares of amplitudes from rejecting states is $\frac{5}{8}$.
- 2. (i=k)and$\neg$(i=j) then the state is
  $\frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}}|q_{a1},0\rangle + \frac{1}{\sqrt{5}}|q_a,0\rangle + \frac{1}{\sqrt{5}}|q_r,0\rangle\right) +$
  $+ \frac{\sqrt{5}}{4}|q_r, j-k\rangle + \frac{\sqrt{6}}{4}|q_a,(j-k)/2\rangle.$
  So the word is accepted with $p=\frac{3}{16}+\frac{1}{16}+\frac{6}{16}=\frac{5}{8}$.
- 3. (j=k)and($\neg$(i=j)). The same as shown in the previous item.
- 4. If all i, j, k are different, then we should distinguish 2 subcases
  - a) if not (i+j)/2=k then word is rejected with probability 1 then word is rejected with probability 1 due to construction of automata - all other transitions from non-halting states to rejecting states.
  - b) if (i+j)/2 =k in this case word is rejected with probability $1-\left(\frac{\sqrt{6}}{4}\right)^2=\frac{5}{8}$.

So the automaton recognizes L_1 with probability $\frac{5}{8}$. Note that this probability is higher then $\frac{3}{5}$ found in [8].

Theorem 2. *The language L_2 can be recognized by quantum one-counter one-way automata with probability 0.58.*

Proof. Let $V_{\neg,0}|q_0\rangle = \frac{1}{\sqrt{5}}|q_{0,i=k}\rangle + \frac{1}{\sqrt{5}}|q_{0,j=k}\rangle + \frac{1}{\sqrt{5}}|q_{0,i=j}\rangle +$

$+\frac{\sqrt{2}}{\sqrt{5}}|q_{0,k=(i+j)/2}\rangle$, where q_0 is initial state, $q_{0,j=k}$, $q_{0,i=k}$, $q_{0,i=j}$, and $q_{0,k=(i-j)/2}$ non-terminating states. Transitions for 0 and 1 can be defined in such way, that it would be reversible and deterministic and the following conditions are satisfied if starting state of such deterministic automaton is $q_{0,j=k}$, $q_{0,i=k}$, $q_{0,i=j}$ or $q_{0,k=(i+j)/2}$ respectively:

1) If the word is in form $0^i10^j10^k$ that no rejection or acceptance occur during the computation and the $q_{0,j=k}$ leads to the state q_{j-k} and counter equal to the j-k, the $q_{0,i=k}$ to $q_{i=k}$ and counter i-k, $q_{0,i=j}$ to $q_{i=j}$ and counter i-j, the $q_{0,k=(i+j)/2}$ to $q_{k=(i+j)/2}$ and counter k - (i+j)/2.
2) If the word is not like $0^i10^j10^k$ than the word is rejected in each path, or is in some other state then $q_{i=j}$, $q_{i=k}$, $q_{j=k}$ and $q_{k=(i+j)/2}$ and thus will be rejected on end marker $\hookleftarrow$.
(Such transitions can be easily defined like shown in proof of Theorem 1).

We define V for $\hookleftarrow$ as
$V_{\hookleftarrow,0}\ |q_{i=k}\rangle = \frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,i=k}\rangle + |q_r\rangle + |q_{a1}\rangle + |q_{a2}\rangle\right)$;
$V_{\hookleftarrow,0}\ |q_{j=k}\rangle =$

$$\frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,j=k}\rangle + |q_r\rangle + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}i\right)|q_{a1}\rangle + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2}i\right)|q_{a2}\rangle\right);$$

$V_{\hookleftarrow,0}\ |q_{i=j}\rangle =$

$$\frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,i=j}\rangle + |q_r\rangle + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2}i\right)|q_{a1}\rangle + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}i\right)|q_{a2}\rangle\right);$$

$V_{\hookleftarrow,0}\ |q_{k=(i+j)/2}\rangle = |q_{r2}\rangle$; $V_{\hookleftarrow,1}\ |q_{k=(i+j)/2}\rangle = |q_{a1}\rangle$,
where $q_{a,i=k}$, $q_{a,i=j}$, $q_{a,j=k}$, q_{a1}, q_{a2} are accepting states and q_r , q_{r2} are rejecting states. All the other transition should be defined to map any of non-halting states to the rejecting states (it can be done by adding some more rejecting states to retain unitarity, see [7])
Let us consider how the computation goes with such automata. Before reading $\hookleftarrow$ word is rejected if it is not of kind $0^i10^j10^k$, and is in the superposition
$|q'\rangle = \frac{1}{\sqrt{5}}\,|q_{i=k}, i-k\rangle + \frac{1}{\sqrt{5}}\,|q_{j=k}, j-k\rangle + \frac{1}{\sqrt{5}}\,|q_{i=j}, i-j\rangle +$
$+\frac{\sqrt{2}}{\sqrt{5}}\,|q_{k=(i+j)/2}, k-(i+j)/2\rangle$.

We should consider then the following cases

- 1. If i=j=k then the state after reading $\hookleftarrow$ due to the sum up of the amplitudes with same states becomes
  $\frac{1}{\sqrt{50}}\left(\sqrt{7}\,|q_{a,j=k}, 0\rangle + \sqrt{7}\,|q_{a,i=k}, 0\rangle + \sqrt{7}\,|q_{a,i=j}, 0\rangle + 3\,|q_r, 0\rangle\right) +$
  $\sqrt{\frac{2}{5}}\,|q_{r2}, 0\rangle$. The total probability of rejection is $\frac{9}{50} + \frac{2}{5} = 0.58$.

- 2. (i=k)and$\neg$(i=j) then the state is
  $\frac{1}{\sqrt{50}}\left(\sqrt{7}\,|q_{a,j=k},0\rangle + |q_{r,},0\rangle + |q_{a1},0\rangle + 3\,|q_{a2},0\rangle\right)+$
  $+\ \frac{1}{\sqrt{5}}\,|q_{r,i=j},i-j\rangle\ +\ \frac{1}{\sqrt{5}}\,|q_{r,j=k},j-k\rangle\ +\ \sqrt{\frac{2}{5}}\,|q_{a},(j-k)/2\rangle.$
  So the word is accepted with p=$\frac{7}{50}+\frac{1}{50}+\frac{1}{50}+\frac{2}{5}$=0.58.
- 3. (j=k)and$\neg$(i=j) The same as shown in the previous item.
- 4. (i=j)and$\neg$(i=k) The same as shown in the previous item.
- 5. If all i, j, k are different, then we should distinguish 2 subcases:
  - a) if not (i+j)/2=k then word is rejected with probability 1 due to construction of automata - all other transitions from non-halting states to rejecting states.
  - b) if (i+j)/2 =k in this case word is rejected with probability 1-$\left(\frac{\sqrt{2}}{5}\right)^2$=0.6.

So the automaton recognizes L_2 with probability 0.58. Note that this probability is higher then $\frac{4}{7}$ found in [8].

References

1. Ambainis, A., and R. Freivalds: 1-way quantum finite automata: strengths, weaknesses and generalizations. Proc. 39^{th} FOCS (1998) 332 – 341
2. Ambainis, A., and J. Watrous: Two-way finite automata with quantum and classical states. http://xxx.lanl.gov/abs/cs.CC/9911009
3. Bonner R., Freivalds R., Kravtsev, M.: Quantum versus Probabilistic Finite One-Counter Automata. In: Proc. 28^{th} SOFSEM (2001), LNCS 2234, Springer-Verlag, 181–190.
4. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. Royal Society London, A400 (1989) 96–117
5. Gruska, J.: Quantum Computing , McGraw Hill (1999)
6. Kondacs, A., and J. Watrous: On the power of quantum finite state automata. Proc. 38^{th} FOCS (1997) 66–75
7. Kravtsev, M.: Quantum Finite One-Counter Automata. In: Proc. 26^{th} SOFSEM (1999), LNCS 1725, Springer-Verlag, 431–440. http://xxx.lanl.gov/abs/quant-ph/9905092
8. Yamasaki, T., Kobayashi H., Tokunaga Y., Imai H.: One-way Probabilistic Reversible and Quantum One-Counter Automata, Proc. COCOON '00 (2000) 436–446
9. Yamasaki, T., Kobayashi H., Imai H.: Quantum Two-Way One-Counter Automata,IMEE 2001. http://xxx.lanl.gov/abs/quant-ph/0110005

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
81–91 P.

Enlarging gap between quantum and deterministic query complexities

Lelde Lāce[1]

Institute of Mathematics and Computer Science University of Latvia
Raiņa bulvāris 29, Riga, LV-1459, Latvia
E-mail address: Lelde.Lace@mii.lu.lv

Abstract Many quantum algorithms can be analyzed in a query (oracle) model where input is given by a black box that answers queries and the complexity of the algorithm is measured by the number of queries to the black box that it uses. It has been proved long ago that quantum query algorithm can compute some Boolean functions with a lesser number of queries. We propose some sets of Boolean functions with leaser complexity.

Keywords. Quantum algorithm, query complexity

1. Introduction

Recently it has become clear that a quantum computer could, in principle, solve certain problems faster than a conventional computer. A quantum computer is a device, which takes full advantage of quantum mechanical superposition and interference. Building an actual quantum computer is probably far off in the future. Boolean decision trees model is the simplest model to compute Boolean functions. In this model the primitive operation made by an algorithm is evaluating an input Boolean variable. The cost of a (deterministic) algorithm is the number of variables it evaluates on a worst-case input.

The *black-box* model of computation arises when one is given a black-box containing an N-tuple of Boolean variables $X=(x_1,x_2,...,x_N)$. The box is equipped to output x_i on input i. We wish to determine some property of X, accessing the x_i only through the black-box. Such a black-box access is called a *query*. A property of X is any Boolean function that depends on X, i.e. a property is function $f:\{0,1\}^N \to \{0,1\}$. We want to compute such properties using as few queries as possible.

Consider, for example, the case where the goal is to determine whether or not X contains at least one 1, so we want to compute the property $OR(X) = x_0 \vee x_1 ... \vee x_{N-1}$. It is well known that the number of queries required to compute OR by any *classical* (deterministic or probabilistic) algorithm is O(N). Grover [4] discovered a remarkable *quantum* algorithm that, making queries in superposition, can be used to compute OR with small error probability using only $O(\sqrt{N})$ queries.

[1] Research supported by Grant No.01.0354 from the Latvian Council of Science and by the European Commission, Contract IST-1999-11234 (QAIP)

2. Definitions

2.1. Quantum computing

We introduce the basic model of quantum computing. For more details, see textbooks by Gruska [5] and Nielsen and Chuang [6]

Quantum states: We consider finite dimensional quantum systems. An n-dimensional pure state is a vector $|\psi\rangle \in C^n$ of norm 1. Let $|0\rangle,|1\rangle,\ldots,|n-1\rangle$ be an orthonormal basis for C^n. Then, any state can be expressed as $|\psi\rangle = \sum_{i=0}^{n-1} a_i |i\rangle$ for some $a_0 \in C$, $a_1 \in C,\ldots,$ $a_{n-1} \in C$. Since the norm of $|\psi\rangle$ is 1, $|a_i|^2=1$. We call the states $|0\rangle,|1\rangle,\ldots,|n-1\rangle$ *basic states*. Any state of the form $\sum_{i=0}^{n-1} a_i |i\rangle$ is called a *superposition* of $|0\rangle,|1\rangle,\ldots,|n-1\rangle$. The coefficient a_i is called *amplitude* of $|i\rangle$.

A quantum system can undergo two basic operations: an unitary evolution and a measurement.

Unitary evolution: A *unitary transformation* U is a linear transformation on C^k that preserves the l_2 norm (i.e., maps vectors of unit norm to vectors of unit norm). If, before applying U, the system was in a state $|\psi\rangle$, then the state after the transformation is $U|\psi\rangle$.

Measurements: In this survey, we just use the simplest case of quantum measurement. It is the full measurement in the computation basis. Performing this measurement on a state $|\psi\rangle = a_1|0\rangle + \ldots a_k|k\rangle$ gives the outcome i with probability $|a_i|^2$. The measurement changes the state of the system to $|i\rangle$. Notice that the measurement destroys the original state $|\psi\rangle$ and repeating the measurement gives the same i with probability 1 (because the state after the first measurement is $|i\rangle$).

More general classes of measurements are general von Neumann and POVM measurements [6].

2.2. Query model

In the query model, the input $x_1,\ldots,x_N$ is contained in a black box and can be accessed by queries to the black box. In each query, we give i to the black box and the black box outputs x_i. The goal is to solve the problem with the minimum number of queries. The classical version of this model is known as *decision trees* [2].

There are two ways how to define the query box in the quantum model. The first is the extension of the classical query (Figure 1).

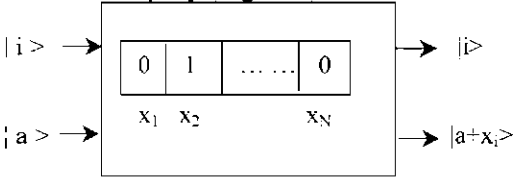


Figure 1. Quantum black box.

It has two inputs i, consisting of [logN] bits and b consisting of 1 bit. If the input to the query box is a basic state $|i\rangle|b\rangle$, the output is $|i\rangle|b\oplus x_i\rangle$. If the input is a superposition $\sum_{i,b} a_{i,b}|i\rangle|b\rangle$, the output is $\sum_{i,b} a_{i,b}|i\rangle|b\oplus x_i\rangle$. Notice that this definition applies both to case when x_i are binary and to the case when they are k-valued. In the k-valued case, we just make b to consist of $\lceil \log_2 k \rceil$ bits and take $b\oplus x_i$ to be bitwise XOR of b and x_i.

In the second form of quantum query (which only applies to problems with $\{0,1\}$-valued x_i), the black box has just one input i. If the input is a state $\sum_i a_i|i\rangle$, the output is $\sum_i (-1)^{x_i} a_i|i\rangle$. While this form is less intuitive, it is very convenient for the use in quantum algorithms, including Grover's search algorithm [4]. A query of second type can be simulated by a query of first type [4].

A quantum query algorithm with T queries is just a sequence of unitary transformations

$$U_0 \rightarrow O \rightarrow U_1 \rightarrow O \rightarrow \dots \rightarrow U_{T-1} \rightarrow O \rightarrow U_T$$

on some finite- dimensional space C^k. U_0, U_1, ..., U_T can be any unitary transformations that do not depend on the bits $x_1, \dots, x_N$ inside the black box. O are query transformations that consist of applying the query box to the first logN+1 bits of the state. That is, we represent basic states of C^k as $|i,b,z\rangle$. Then, O maps $|i,b,z\rangle$ to $|i,b\oplus x_i,z\rangle$. We use O_x to denote the query transformation corresponding to an input $x=(x_1,\dots x_N)$.

The computation starts with state $|0\rangle$. Then, we apply U_0, O_x ,..., O_x, U_T and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement (or several bits if we are considering a problem where the answer has more than 2 values).

The quantum algorithm computes a function $f(x_1,\dots,x_N)$ if, for every $x=(x_1,\dots,x_N)$ for which f is defined, the probability that the rightmost bit of $U_T O_x U_{T-1} \dots O_x U_0|0\rangle$ equals $f(x_1,\dots,x_N)$ is at least $1-\varepsilon < \frac{1}{2}$.

The exact quantum algorithm computes a function $f(x_1,\dots,x_N)$ if, for every $x=(x_1,\dots,x_N)$ for which f is defined, the probability that the rightmost bit of $U_T O_x U_{T-1} \dots O_x U_0|0\rangle$ equals $f(x_1,\dots,x_N)$ is 1.

The query complexity of f is the smallest number of queries used by a quantum algorithm that computes f. We denote it $Q(f)$.

3. Main results

3.1. Constructions

First construction is Hadamard transformation (Figure 2). This is a unitary transformation.

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Figure 2. Hadamard transformation.

In quantum algorithms Hadamard transformation is used like this (Figure 3).

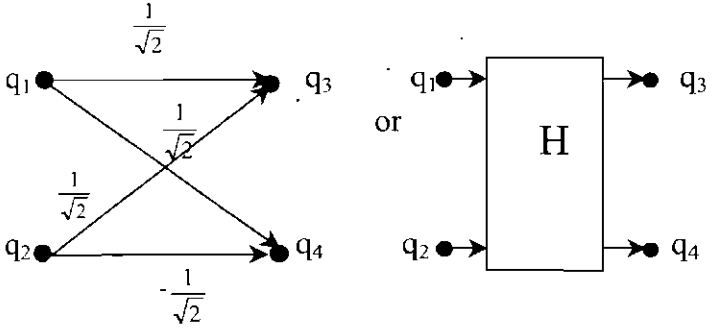


Figure 3. Hadamard transformation in quantum algorithm.

Distribution of amplitudes (before and after applying Hadamard transformation) is shown in Table 1.

	(q_1,q_2)	(q_3,q_4)
1	$\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$	(1,0)
2	$\left(\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\right)$	(0,1)
3	$\left(-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$	(0,-1)
4	$\left(-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\right)$	(-1,0)

Table 1. Distributions of amplitudes.

Next construction solves the following problem. We have one variable x, after a query we get amplitude 1 in state q_2, if value of x is 1 and we get amplitude -1 in state q_2, if value of x is 0. Algorithm, which corresponds requirements, is shown in Figure4.

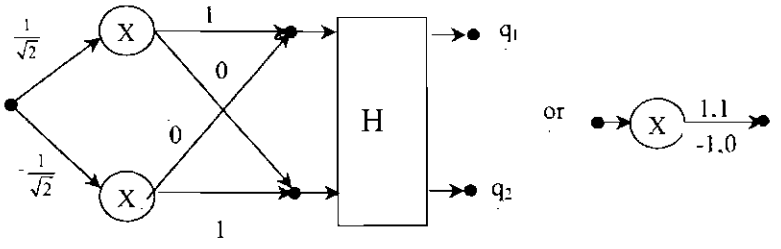


Figure 4. Algorithm

D.Deuch [7] gave a construction (finalized by R.Cleve, A.Ekert. C. Macchiavello and M.Mosca [8]) computing the Boolean function $PARITY(x_1,x_2)=x_1+x_2(\text{mod } 2)$.

Algorithm, which computes *PARTY* with probability 1 (exact quantum query Algorithm), is shown in Figure 5.

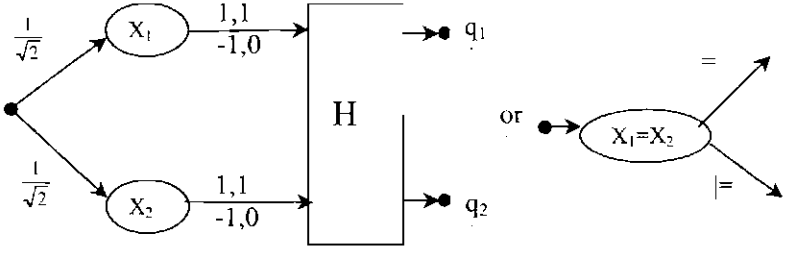


Figure 5. Automaton, which calculate *PARITY*

Distribution of amplitudes (algorithm, which computes *PARITY*) is shown in Table 2.

x_1,x_2	After query	(q_1,q_2)
(1,1)	$\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$	(1,0)
(1,0)	$\left(\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\right)$	(0,1)
(0,1)	$\left(-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$	(0,-1)
(0,0)	$\left(-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\right)$	(-1,0)

Table 2. Distributions of amplitudes.

Theorem 1 *Let Q be an exact quantum query algorithm with two outputs (0,1). This algorithm computes Boolean function $f_1(x_1,...x_n)$ with k_1 queries. Corresponding deterministic algorithm requires k_2 queries ($k_2>k_1$). Let D be a deterministic reversible query algorithm, which computes Boolean function $f_2(x_1,...x_n)$ with n queries.*

Then exists exact quantum query algorithm Q_2, which computes function $f_2(f_1(x_1,...x_m),f_1(x_{m+1},...x_{2m}),...f_1(x_{(n-1)m+1},...x_{nm}))$ with k_1n queries and corresponding deterministic query algorithm have k_2n queries.

Proof. We construct algorithm Q_2 as follows. We put in each deterministic algorithm D query quantum query algorithm Q with corresponding variables. In this way we get correct quantum query algorithm, which computes Boolean function f_2 with k_1n queries. Corresponding deterministic algorithm is made in a similar way. This algorithm requires k_2n queries. □

Theorem 2 *Let $f(x_1,...x_n)$ be a Boolean function. If if $x_1=x_2=...x_n=0$ then value of $f(x_1,...x_n)$ is 1. If one of x_i is 1, but others are 0 then value of $f(x_1,...x_n)$ is 0.*

Then deterministic query algorithm requires n queries.

Proof. If we ask n-1 queries and all values are 0, then we need to ask another query to find out value of function f. □

3.2. Exact quantum query algorithm with n-1 queries

In this chapter we propose set of Boolean functions, which complexity of exact quantum query algorithm is n-1 but complexity of deterministic query algorithm is n.

The idea about differences between two input variables has come from Raitis Ozols.

Definition *Let $x_1,... ,x_n$ be a Boolean functions input. Then difference* ***Dif(i)*** *between two input variables x_i and x_{i-1} is:*

$$\underset{i=1,n}{Dif}(i)=\begin{cases}0, x_i = x_{i-1(\mathrm{mod}\, n)}\\ 1, x_i \neq x_{i-1(\mathrm{mod}\, n)}\end{cases}$$

Definition *Let $x_1,... ,x_n$ be a Boolean functions input and Dif(i) corresponding differences, then*

$$DifSum(x_1,...x_n)=\sum_{i=1,n} Dif(i)$$

Theorem 3 [9] *If $x_1,... ,x_n$ is Boolean functions input and $n=2k-1$, then DifSum $\in$ {0,2,4..,n-1}.*

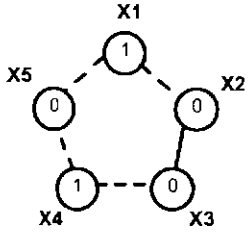


Figure 6. Example of five input variables

Example of five input variables is shown in Figure 6. If the difference between two contiguous variables is 0 (variables are equal), then in figure this fact is shown with bold line. If the difference between two contiguous variables is 1 (variables are not equal), then in figure this fact is shown with discontinuous line.

Exact query algorithm, which can differentiate different values of *DifSum*, is shown in Figure 7. This algorithm has four queries, as each confrontation of two variables requires only one query.

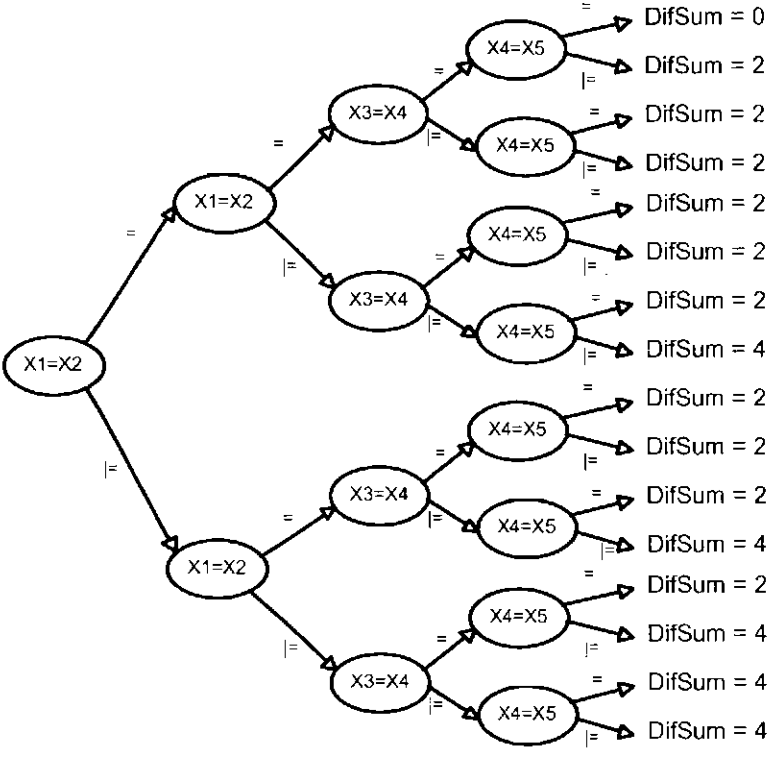


Figure 7. Quantum query algorithm

We consider Boolean function of n variables

$$H_1(x_1,\ldots x_n)=\begin{cases}1, DifSum(x_1,\ldots x_n)=0\\ 0, DifSum(x_1,\ldots x_n)>0\end{cases}$$

Theorem 4 *There is an exact quantum query algorithm for Boolean function H_1 with n-1 queries and deterministic query algorithm requires n queries.*

Proof. We make this quantum query algorithm alike five variables situation. The algorithm has n-1 queries. That output, which corresponds DifSum=0, gives value 1, other outputs give value 0. This algorithm computes Boolean function H_1. Boolean function H_1 satisfies requirements of Theorem 2, so deterministic query algorithm requires n queries. □

Theorem 5 *Let H be a Boolean function of n variables and*

- *H value is 1, if $DifSum(x_1,\ldots x_n)=0$*
- *H value is 0, if $DifSum(x_1,\ldots x_n)=2$*
- *For any two different inputs H value is the same, if value of inputs DifSum is the same.*

Then there is an exact quantum query algorithm for Boolean function H with n-1 queries and deterministic query algorithm requires n queries.

Proof. We make this quantum query algorithm alike Theorem 4. We have Boolean function H, which value is dependent only on input DifSum value. This means, we can get necessary outputs from the quantum query algorithm. Boolean function H satisfies requirements of Theorem 2, so deterministic query algorithm requires n queries. □

3.3. Exact quantum query algorithm with 2n/3 queries

In this chapter we propose set of Boolean functions, which complexity of exact quantum query algorithm is 2n/3 but complexity of deterministic query algorithm is n.

We consider Boolean function of 3 variables

$$G(x_1,x_2,x_3)=\begin{cases}1, x_1=x_2=x_3\\ 0, otherwise\end{cases}$$

Theorem 6 *There is an exact quantum query algorithm Q_G for G with 2 queries and this algorithm has two outputs.*

Proof. Quantum query algorithm shown in Figure 8 satisfies requirements. Output q_1 corresponds G value 1 and output q_2 -G value 0. □

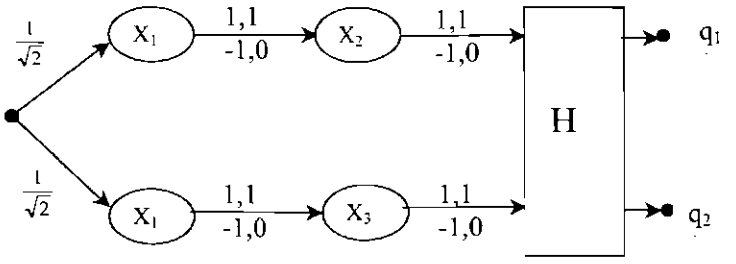


Figure 8. Quantum query algorithm Q_G

Theorem 7 *There is a set of Boolean functions, which deterministic complexity is 3n and exact quantum query algorithm with complexity 2n.*

Proof. We use Theorem 1 and Theorem 2. Quantum query algorithm Q_G (from Theorem 6) satisfies requirements of Theorem 2. We take any Boolean function d of n variables, which satisfies requirements of Theorem 2. This means the function has deterministic query algorithm D with n queries. Then by Theorem1, we can make exact quantum query algorithm, which computes function $d(g(x_1,x_2,x_3),\ldots,g(x_{3n-2}, x_{3n-1}, x_{3n}))$ with 2n queries. By Theorem 2 corresponding deterministic quantum algorithm requires 3n queries. □

3.4. Exact quantum query algorithm with n/2 queries

In this chapter we propose set of Boolean functions, which complexity of exact quantum query algorithm is n/2 but complexity of deterministic query algorithm is n.

We consider Boolean function of 4 variables

$F(x_1,x_2,x_3,x_4)=PARITY(x_1,x_2,x_3,x_4)$

Theorem 8 *There is an exact quantum query algorithm Q_F for F with 2 queries and this algorithm has two outputs.*

Proof. Quantum query algorithm shown in Figure 9 satisfies requirements. Output q_1 corresponds F value 1 and output q_2 - F value 0. □

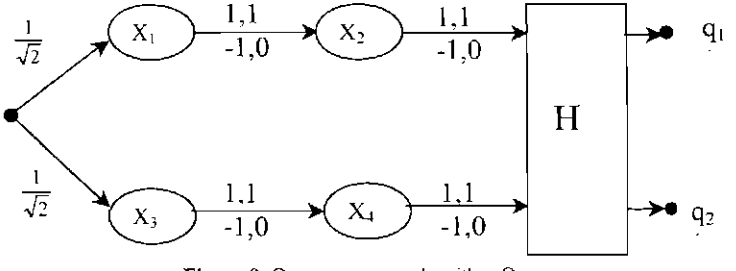


Figure 9. Quantum query algorithm Q_F

Theorem 9 *There is a set of Boolean functions, which deterministic complexity is 4n and exact quantum query algorithm with complexity 2n.*

Proof. We use Theorem 1 and Theorem 2. Quantum query algorithm Q_F (from Theorem 8) satisfies requirements of Theorem 2. We take any Boolean function d of n variables, which satisfies requirements of Theorem 2. This means the function has deterministic query algorithm D with n queries. Then by Theorem1, we can make exact quantum query algorithm, which computes function $d(f(x_1,x_2,x_3,x_4),\ldots, f(x_{4n-3},\ldots, x_{4n}))$ with 2n queries. By Theorem 2 corresponding deterministic quantum algorithm requires 4n queries. □

3.5. Exact quantum query algorithm with 2 queries for Boolean function of 4 variables

In this chapter we propose Boolean function of 4 variables, which complexity of exact quantum query algorithm is 2 but complexity of deterministic query algorithm is 4. This function do not equivalent with PARITY function.

We consider Boolean function of 4 variables.

$$K(x_1,x_2,x_3,x_4) = \begin{cases} 1, DifSum(x_1,\ldots x_4) = 0 \\ 1, DifSum(x_1,\ldots x_4) = 4 \\ 0, DifSum(x_1,\ldots x_4) = 2 \end{cases}$$

Theorem 10 *There is an exact quantum query algorithm Q_K for K with 2 queries.*

Proof. Quantum query algorithm shown in Figure 10 satisfies requirements. Output q_1 corresponds K value 1 and outputs q_2, q_3, and q_4 - K value 0. □

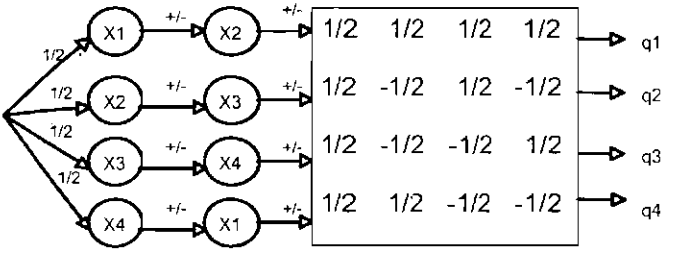


Figure 10. Quantum query algorithm Q_K

References

[1] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, Umesh V. Vazirani. Strengths and Weaknesses of Quantum Computing. SIAM Journal on Computing, v. 26, 1997, pp. 1510 1523.

[2] H. Buhrman and R. de Wolf. Complexity Measures and Decision Tree Complexity : A Survey. Theoretical Computer Science, v. 288(1): 21-43 (2002)

[3] Rūsiņš Freivalds, Andreas Winter: Quantum Finite State Transducers. SOFSEM 2001: 233-242

[4] L. Grover. A fast quantum mechanical algorithm for database search. Proceedings of the 28th ACM symposium on Theory of Computing, pp 212-219, 1996.

[5] J. Gruska. Quantum Computing. McGraw-Hill, 1999.

[6] M. Nielsen, I. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000

[7] David Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society, London, A400:97-117, (1985).

[8] Richard Cleve, Arthur Ekert, Chiara Macchiavello, Michele Mosca. Quantum Algorithms Revisited. Proceedings of the Royal Society London, A454 (1998) , p. 339-354.

[9] Raitis Ozols. Personal communication. 2004.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673. Databases and Information Systems
92–104 P.

Metamodels and Formalization of Fuzzy Knowledge: a Case Study

Dace Rukliša, Jānis Bārzdiņš

University of Latvia. IMCS
29 Raina boulevard, Riga, Latvia
{dace.ruklisa, janis.barzdins}@mii.lu.lv

Abstract. The paper focuses on study of a fuzzy concept of easily inferred sequences introduced by D.Angluin. A method for formalization of such fuzzy concepts is developed. The proposed method is based on object-modeling technique. Using this method a metamodel that adequately describes easily inferred sequences is constructed.

Keywords: Easily inferred sequences, inference patterns, metamodels, object-modeling technique

1. Introduction

Current paper has several goals. The first goal is to develop an algorithm for inference of easily inferred sequences. First informal definition of easily inferred sequences was introduced by D.Angluin 30 years ago [1]. Easily inferred sequence is a sequence of natural numbers; a human easily guesses the correct rule of generation for such a sequence from a few successive terms. The authors of this paper have not found a method of computerized inference that could compete with humans in the efficiency up to now. Only some classes of easily inferred sequences are deeply studied. An interesting attempt to formalize several sequence classes together with their inference algorithm is the synthesis of dot expressions [4]. Inductive inference of expressions (including expressions used in the generation of sequences) from input/output examples is studied in [2]. However, there is no method of inference that can deal with the whole variety of sequences. One of the reasons for it might be the lack of formal description of such sequences.

The second goal is to develop a general method for formalization of fuzzy templates. Humans use such templates when they induce general knowledge from specific examples. Several problems should be solved in order to develop a method of formalization. The first problem is how to chose and describe elementary templates. The second problem is how to combine elementary templates to obtain more powerful tools of formalization. The problem of combining them in a formal way is non-trivial, because templates of inference used by humans usually are orthogonal. Since the 16th century we

know how to combine arithmetic operations to obtain so-called algebraic expression. However, as a tool of formalization of sequences we need more general notions than the notion of algebraic expression.

The next goal is to use the object-modeling technique ([5], [3]) in formalization of easily inferred sequences. The object-modeling technique emerged as a description method of complex systems in a language that is readable and perceivable by a human. Traditionally it is used in the formalization of the IT domains – banks, insurance companies, telecom, sophisticated program systems. In this work we will study how the object-modeling technique can be applied to describe combinations of orthogonal patterns.

Traditional programming languages are not suitable for formalization of easily inferred sequences. It is hard to imagine a sieve that could efficiently divide programs into "easily inferable" and "hard inferable". It is a great minus also in inference of sequences. As the count of all programs of fixed length is exponential, it might be difficult to avoid the exponential search over programs that generate sequences.

It is important to find a description form of sequences where notions that are familiar to and understandable for a human are used. Besides, the depth of template compositions in the description of a sequence should not exceed the path of inference characteristic for a human. If we find such form of description that satisfies these requirements, it would be possible to carry out fast enough exhaustive search over generation templates of easily inferred sequences. It would be beneficial to supplement this search also with some heuristics.

2. Easily inferred sequences: examples

In this section different examples of easily inferred sequences will be shown. Of course, these examples are only a small portion of the overall variety of easily inferred sequences. In D.Angluin's paper approximately 60 templates of sequences are described [1]. Each template can be used to generate many sequence examples. Besides, these templates can be combined to obtain more complex templates and thus even more examples of sequences.

1, 4, 9, 16, 25, 36 ...

The first sequence is the sequence of squares. Each element is equal to the square of its number in the sequence.

7, 10, 13, 16, 19, 22 ...

This sequence is an arithmetic progression with difference 3.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ...

This is a Fibonacci sequence.

1, 1, 2, 8, 3, 27, 4, 64, 5, 125, 6 ...

This is an alternating sequence that consists of two subsequences: 1, 2, 3, 4, 5, 6 ... and 1, 8, 27, 64, 125 ... . Both subsequences are intertwined so that elements in even places are cubes of the previous element in odd place.

1, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, 8, 16 ...

This sequence is made of a sequence of powers of 2: 1, 2, 4, 8, 16 ... . Each power is repeated as many times as its value.

1, 0, 4, 0, 9, 0, 1, 6, 0, 2, 5, 0, 3, 6, 0, 4, 9 ...

The backbone of this sequence is the sequence of squares 1, 4, 9, 16, 25, 36, 49 ... . A 0 is put between every two successive elements of square sequence. Finally, each element is divided into separate digits.

From these examples we can see that the templates of sequence generation are really orthogonal. At the same time combinations of orthogonal templates in sequences are still easily inferable for a human.

3. Introductory ideas of formalization

The basic element of formalization of easily inferred sequences will be an elementary pattern. Elementary patterns will be presented as the class diagrams of the form similar to Fig.1.

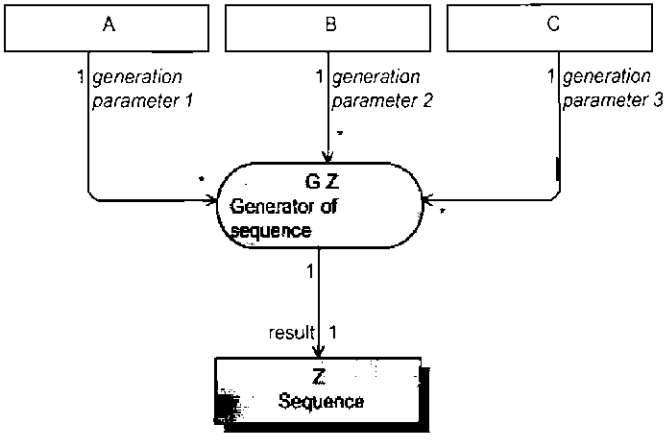


Fig. 1. Representation of elementary pattern

Class diagrams of elementary patterns use stereotypes depicted in Fig.2.

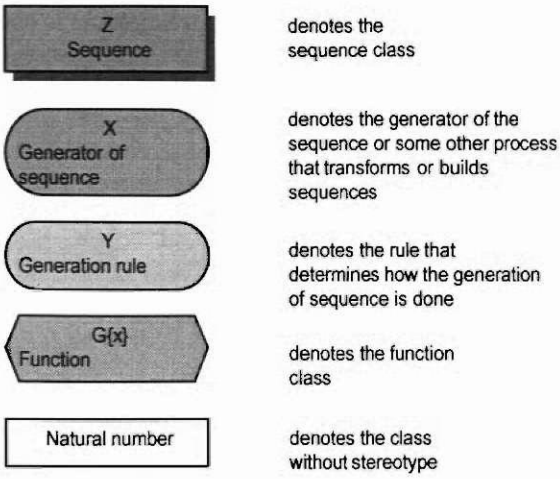


Fig. 2. Stereotypes

Pattern in Fig.1. shows the generation of sequences from the class Z. The generation is done by the generator G_Z. Generator takes several generation parameters A, B, C and produces a sequence from the class Z. Classes A, B, C commonly stand for some sequence classes or function classes. The roles *generation parameter 1*, *generation parameter 2* and *generation parameter 3* can be considered as role stereotypes. These stereotypes will be substituted by concrete roles when describing particular patterns of easily inferred sequences.

Let us consider an example of sequence generation pattern (Fig.3).

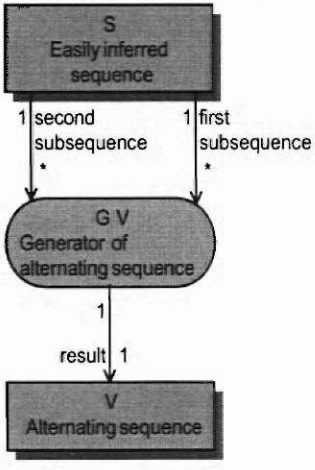


Fig. 3. Example of sequence generation pattern

This class diagram depicts the alternation of simple sequences to obtain an alternating sequence. The generator G_V takes the necessary generation parameters and produces an alternating sequence. Parameters include two simple sequences. In the process of generation odd elements of the alternating sequence are taken from the first subsequence, but even elements are obtained from the second subsequence.

Let us build an instance diagram of the alternating sequence 1, 2, 2, 4, 3, 8, 4, 16, 5, 32, 6, 64 ... (Fig.4).

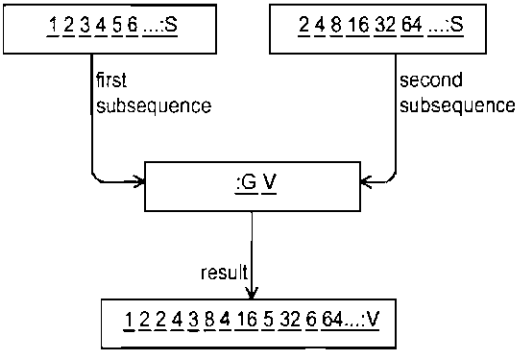


Fig. 4. Example of instance diagram

In this paper rules of sequence generation will be described in natural language. Rules of generation will be comparatively simple for elementary patterns.

Various functions are widely used as generation parameters. Here we will define some classes of easily inferable functions.

F(x) : arithmetic function with one occurrence of a variable, some occurrences of natural constants and arithmetic operators +, -, *, ^. Total amount of operator occurrences does not exceed 4 (otherwise function may become hard inferable).

F(x, y) : arithmetic function that is of form x+y or x*y.

S(x) : decimal function with one operator, which works with decimal representation of a natural number. Operator can be of the type *firstdigit*, *lastdigit*, *length*, *tail*, *front*.

Let us define these decimal operators.

Firstdigit : returns the first digit in the decimal notation of a natural number; *firstdigit*(81) = 8.

Lastdigit : returns the last digit in the decimal notation of a natural number; *lastdigit*(127) = 7.

Length : returns the number of digits in the decimal notation of a natural number; *length*(25) = 2.

Tail : returns the number without the first digit in the decimal notation; *tail*(243) = 43.

Front : returns the number without the last digit in the decimal notation; *front*(1024) = 102.

R(x) : this type of functions includes arithmetic functions like F(x) and functions of form *front*(x) and *tail*(x).

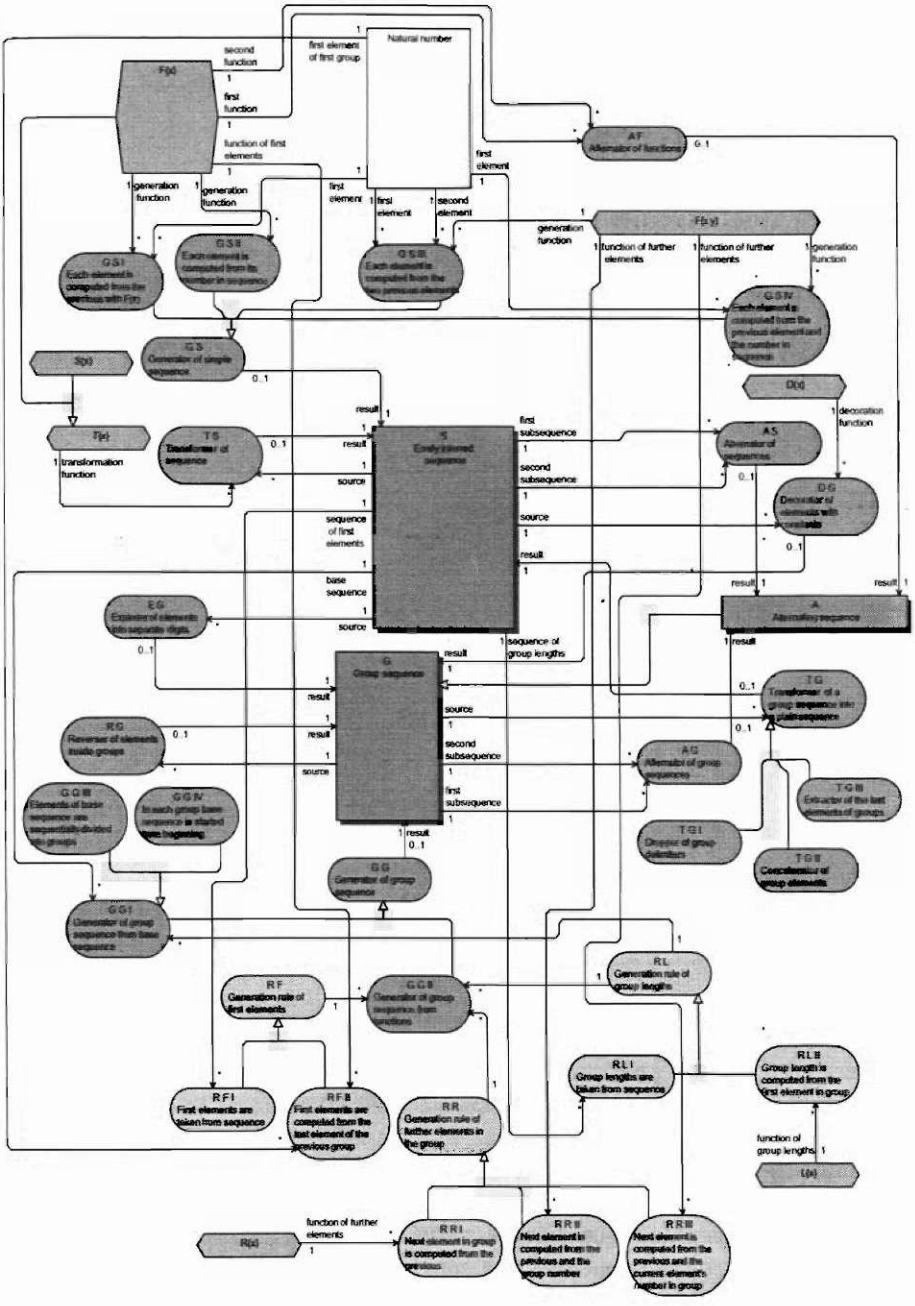


Fig. 5. Metamodel of easily inferred sequences

L(x) : class L(x) includes arithmetic functions F(x) and functions of form *length*(F(x)).

D(x) : function D(x) returns the list $<c_1, c_2, \ldots x, \ldots c_n>$, where $c_1, c_2, \ldots c_n$ are fixed constants and x is an occurrence of a variable.

4. Metamodel of easily inferred sequences

The metamodel of easily inferred sequences is depicted in Fig.5. This metamodel contains elementary patterns and their possible combinations. From formal point of view every easily inferred sequence is defined as an instance diagram of the given metamodel.

A typical instance diagram of easily inferred sequence is depicted in Fig.6. This diagram shows the generation of the sequence 1, 0, 4, 0, 9, 0, 1, 6, 0, 2, 5, 0, 3, 6, 0, 4, 9 ... .

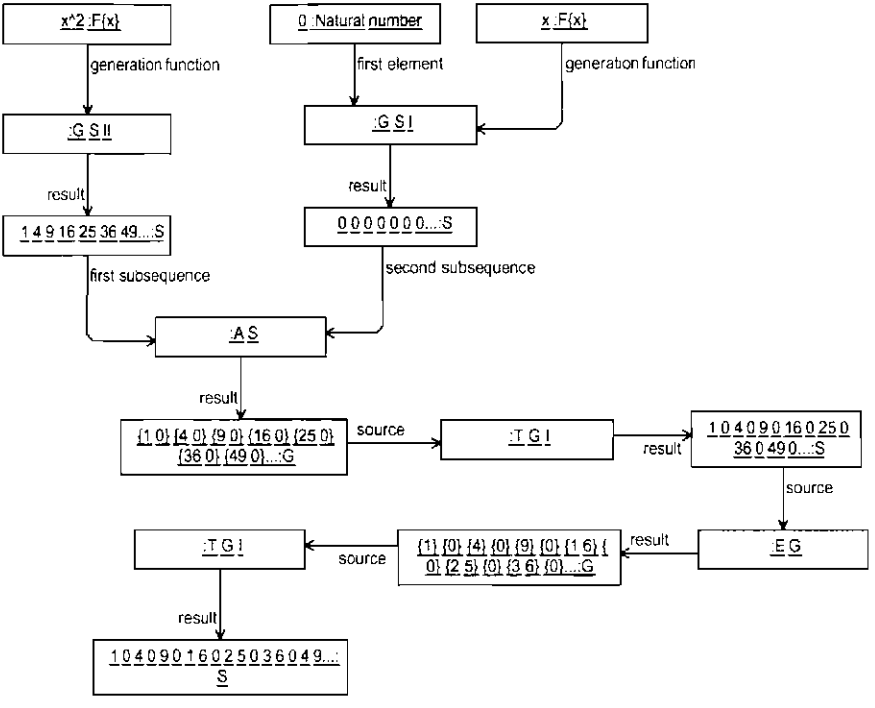


Fig. 6. Instance diagram of easily inferred sequence

The semantics of the metamodel will be discussed in further sections. For the sake of explanation the metamodel will be divided into several fragments that unite similar patterns. Patterns will be described in a greater detail in the context of these fragments.

4.1. Simple generation patterns

Formalization of easily inferred sequences is started with some simple patterns (Fig.7).

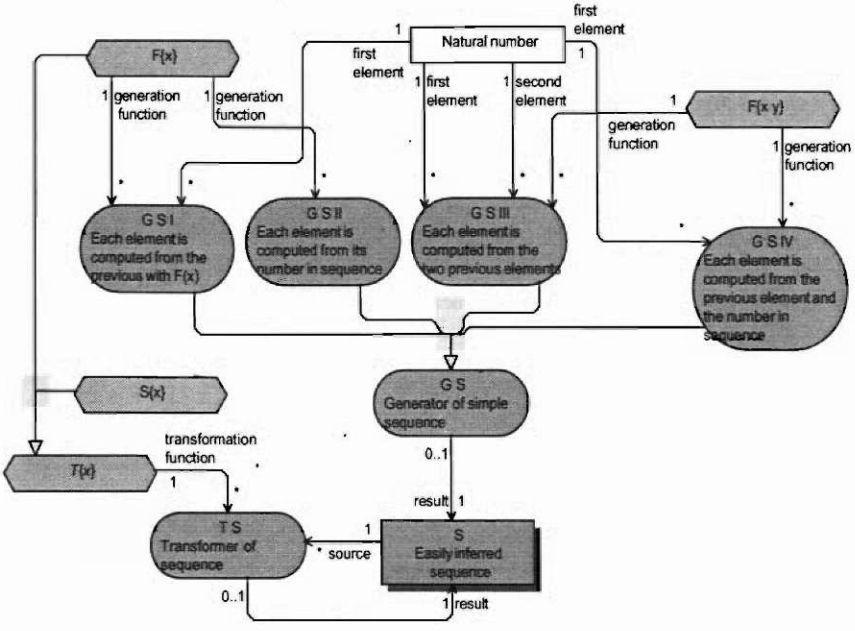


Fig. 7. Simple generation patterns

The class S represents an easily inferred sequence. There are four generators of this sequence class (from G_S_I to G_S_IV). The first generator G_S_I uses the function F(x) and natural number as generation parameters. During generation every element of the sequence, except the first one, is obtained from the previous element by applying function F(x). The first element is stated explicitly as natural number. The generator G_S_II uses the function F(x) to compute elements in the sequence from their numbers in the sequence. The generation template G_S_III allows the generation of sequences similar to Fibonacci sequence. In these sequences next element is computed from the two previous elements. The template G_S_IV describes sequences where next element is computed from the previous element and the number of current element in the sequence.

If we have some easily inferred sequence already, it can be transformed into another sequence. Transformation means that every element x_i of the source sequence is transformed into element $f(x_i)$, where f is the transformation function. The transformer T_S can operate with two types of functions – arithmetic and decimal functions.

In the following sections we will see yet other methods of producing an easily inferred sequence.

4.2. Group sequence

Now we will consider a more general type of sequence – the group sequence. The group sequence is a sequence built from finite subsequences called groups. Every finite subsequence is a fragment of some easily inferred sequence. If these subsequences together form some pattern, then the resulting group sequence also is easily inferred. Group lengths should form some simple pattern as well.

Let us assume that each group is delimited from other groups by brackets. Thus, for each group sequence there is a fixed partition of elements into groups. For example, group sequence 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5 ... in our inner representation might look as follows <1>, <2, 2>, <3, 3, 3>, <4, 4, 4, 4>, <5 ...> ... .

First, we will look at some patterns of transformation of plain sequences (easily inferred sequences) into group sequences, and group sequences into plain sequences (Fig.8). Then we will pass over to more complex generation patterns of group sequences (Fig.9).

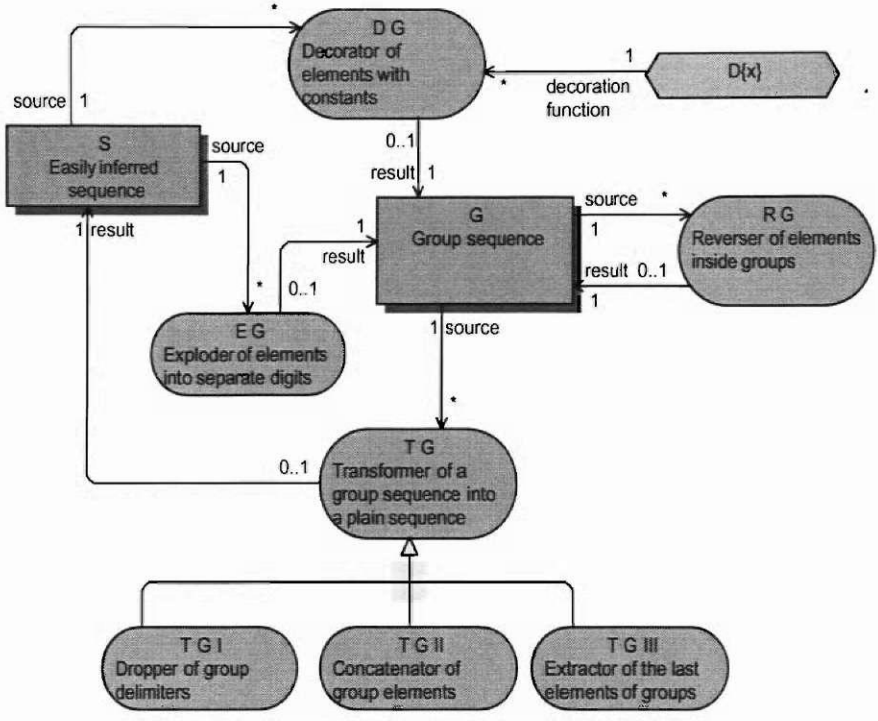


Fig. 8. Transformation of group sequence

The group sequence corresponds to the class G in the metamodel.

An easily inferred sequence can be transformed into a group sequence either by decorator D_G or exploder of the elements of sequence E_G. The decoration is done by

the function D(x), which takes a natural number and returns a list of numbers, where constants are put around the argument x. D(x) is applied to each element x_i in the source sequence; the returned list $D(x_i)$ constitutes one group. The exploder E_G takes each element and converts it into a list of separate digits. The list of digits obtained from one element of the source sequence forms one group.

The reverser R_G can transform group sequence into another group sequence. The reverser translates each group of the source sequence into another group, where elements are put in a reverse order.

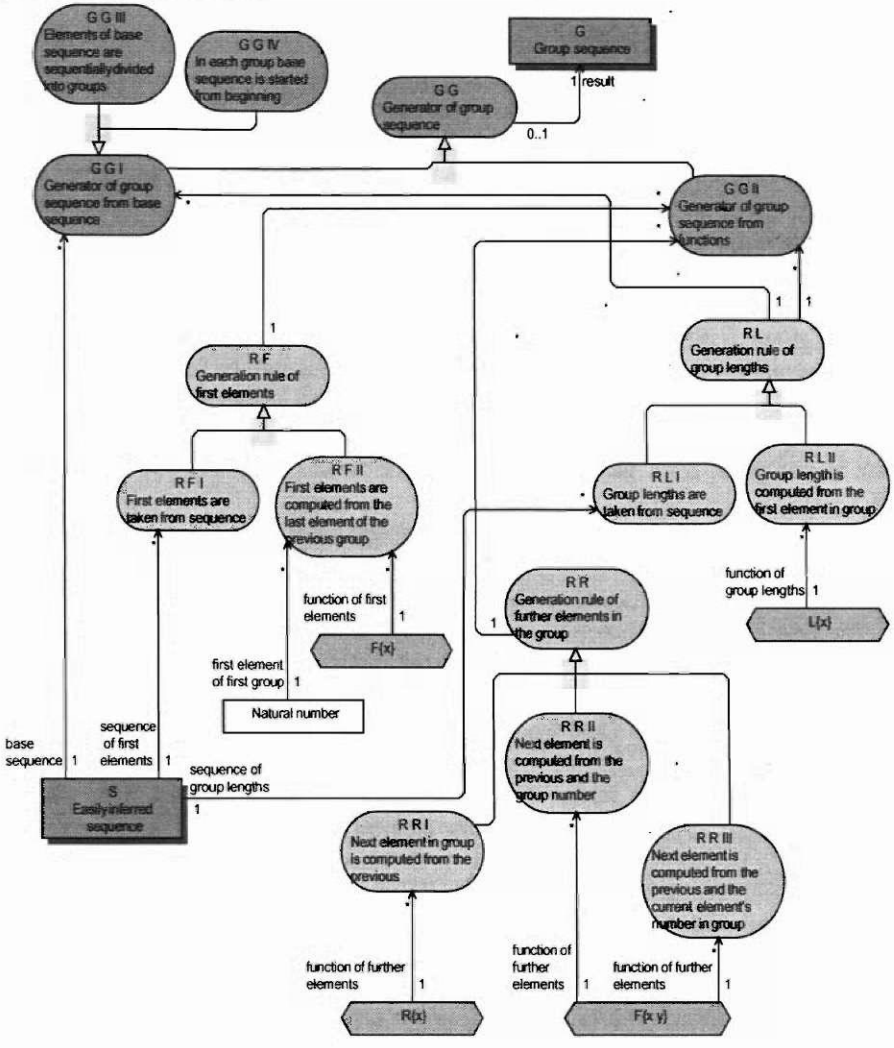


Fig. 9. Generation of group sequence

There are three ways how to construct a plain sequence from a group sequence (the transformers T_G_I, T_G_II and T_G_III in Fig.8). First, we can drop brackets of groups to obtain a plain sequence (the transformer T_G_I). The transformer T_G_II concatenates elements inside groups and makes a sequence, where each number is equal to concatenation of some group. The transformer T_G_III makes a sequence that contains last elements of groups in the group sequence.

Now we are ready to pass to generation of group sequences (Fig.9). The generation of a group sequence can be done in two ways: using base sequence as a source of group elements (G_G_I) or using functions to derive groups (G_G_II).

Two parameters are involved in the generation of a group sequence that uses base sequence: the base sequence and the pattern of group lengths. The base sequence can be used in two different ways (represented by generators G_G_III and G_G_IV). First, the elements of the sequence can be sequentially divided into groups. It means that the first l_1 elements constitute the first group, next l_2 elements form the second group etc., where l_1 is the length of the first group. l_2 is the length of the second group and so on. Another way of forming a group sequence is to repeat the base sequence in each group. In this case the first group consists of the first l_1 elements of the base sequence, the second group consists of the first l_2 elements of the base sequence etc., where l_1 is the length of the first group, l_2 is the length of the second group and so on.

The group length specifies how many natural numbers will be in a group. There are two possible sources of group lengths according to the generation rule R_L. Group lengths can be given as elements of another easily inferable sequence (R_L_I), or they can be computed from the first element of the same group with the function L(x) (R_L_II).

The generation of a group sequence from functions is based on three fundamental rules: the generation rule of the first elements of groups (R_F), the derivation rule of further elements in the group (R_R) and the generation rule of the group lengths (R_L).

The first elements of groups may correspond to elements of some easily inferable sequence (rule R_F_I), or they can be computed from last element of the previous group with the function F(x) (rule R_F_II). In the second case the generation rule requires two parameters: natural number and function. The natural number serves as the first element of the first group.

The generation rule of further elements specifies the generation of group elements starting from the second, assuming the first element is already computed. According to the rule R_R_I next element in the group is obtained from the previous element in the group by applying function R(x). The rule R_R_II uses a two-argument generation function to return next element in the group. The first argument of this function is the previous element in the group, but the second argument is the group number. The rule R_R_III uses the same kind of function as the rule R_R_II, but the second argument of function is the number of the current element inside the group.

Group lengths can be taken either from another sequence or computed from the first element of the same group.

4.3. Alternating sequence

The last type of easily inferred sequences in our classification is the alternating sequence. The alternating sequence is a composition of two subsequences or two functions. The structure of an alternating sequence that is made of two subsequences can be depicted as follows:

<1-st element of 1-st subsequence,
1-st element of 2-nd subsequence>,
<2-nd element of 1-st subsequence,
2-nd element of 2-nd subsequence>,
...
<k-th element of 1-st subsequence,
k-th element of 2-nd subsequence>,
...

An alternating sequence of two functions has the following structure:

<1-st element,
f(1-st element)>,
<g(2-nd element),
f(3-rd element)>,
...
<g(2k-th element),
f(2k+1-th element)>,
...

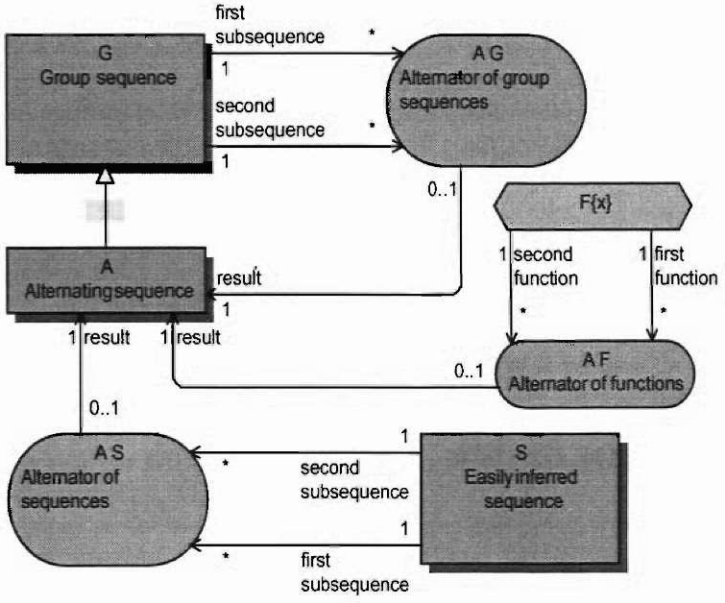


Fig. 10. Alternating sequence

In this model the alternating sequence is a subclass of the group sequence (Fig.10). General structure of the alternating sequence indicates that elements are grouped in pairs. Either neighbouring elements from different subsequences are paired, or neighbouring results of different functions form one pair. When the alternating sequence is made of two group sequences, the delimiters of groups in subsequences are dropped. For example, alternating sequence made of sequences <1>, <2, 2>, <3, 3, 3>, <4, 4, 4, 4> ... and <1>, <1, 2>, <1, 2, 3>, <1, 2, 3, 4> ... with the alternator A_G is <1, 1>, <2, 2, 1, 2>, <3, 3, 3, 1, 2, 3>, <4, 4, 4, 4, 1, 2, 3, 4> ... .

5. Conclusion

In this paper we have demonstrated the method of sequence formalization. For the sake of shortness we have chosen quite a small set of elementary patterns. Due to small amount of patterns some sequences are depicted as a composition of several patterns, even though a human perceives them as a whole. Therefore in the final version we have included additional patterns. Additional patterns simplify the representation of sequences, but are more complex.

Practically all easily inferred sequences can be depicted as instances of the class diagram in Fig.5. In this sense our set of elementary patterns is full.

Further aim of this work is effective implementation of search over instance diagrams. The best measurement of the effectiveness of search would be the competition with a human.

References

[1] Angluin D. Easily Inferred Sequences. Memorandum No. ERL-M499, University of California, Berkeley, 1974, 25 pp.

[2] Bārzdiņš J., Bārzdiņš G., Apsītis K., Sarkans U. Towards Efficient Inductive Synthesis of Expressions from Input/Output Examples. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 1993, vol.744, pp. 59-72.

[3] Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, Addison-Wesley, Reading Harlow, 1999, 512 pp.

[4] Brāzma A. Inductive Synthesis of Dot Expressions. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 1991, vol.502, pp. 156-212.

[5] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. Object-Oriented Modeling and Design, Prentice Hall, 1991, 500 pp.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673. Databases and Information Systems
105–114 P.

Multi-Indices - A Tool for Optimizing Join Processing in Main Memory

Dmitry Shaporenkov

University of Saint-Petersburg, Russia
dsha@acm.org

Abstract. In this paper we revise a well-known technique for optimizing joins - join-indices. We propose a variant of join-indices, multi-indices, which are specifically tailored for main-memory databases. We discuss trade-offs for creating multi-indices, outline implication of multi-indices on update and insert procedures and describe their usage in query processing algorithms. For some important particular kinds of queries involving selections we also propose a further optimization that places a pointer to the shared index record into the data record and thereby avoids search in the index structure altogether.

Keywords. Access methods, join algorithms, main-memory databases

1 Introduction

In the last decade advances in computer hardware allowed to store relatively large databases in main memory, attracting great attention to main-memory databases in research community. It has been shown that main-memory databases provide performance on order of magnitude better than traditional, disk-based databases [3]. A main-memory database system (MMDBMS) can store all the data and support structures (such as indices) in main memory, using disk only for logging and recovery and avoiding inefficient random access to the mechanical device [2].

Changing primary data storage from disk to main memory does not free database systems developers from the problems peculiar to traditional databases. In particular, efficient data access is still very important for MMDBMS. However, MMDBMS bring new criteria for evaluating access methods - CPU cache utilization. Recent research has shown that CPU cache misses are the biggest performance bottleneck for MMDBMS [7, 1]. Many algorithms and data structures traditionally used in DBMS have been revised in last years from the viewpoint of cache behavior [8, 5, 11, 1].

In this paper we focus on performance of equi-join operation, which is widely used in query processing algorithms for relational DBMS. We propose *multi-indices* - a sort of indices mapping an attribute value to all records of different relations containing this value. Multi-indices employ the same idea as the well-known join indices - precomputing information required during join operation. Using multi-index, one can find all pairs of matching records without scanning

or preprocessing relations. We propose a layout of index structures that seems to be optimal with respect to cache behavior. For queries involving selections, performance can be further improved by placing a pointer to the index record into the data record. We expect this modification to significantly speed up execution of such queries.

2 Related work

The CSB+-tree proposed in [8] features a layout of tree nodes with only one child pointer. This structure frees extra space in a node for keys, increasing tree branching factor and decreasing height of the tree. This reduces number of cache misses during tree traversal at the cost of more complex node split algorithm. One of the latest works in the field, [11], uses a buffering technique for optimizing search in B+-tree. Buffering helps to avoid cache miss occurring when walking down from a parent to a child in the tree. An algorithm is described that distributes buffers among the tree nodes during query processing thus accommodating buffering strategy to the workload.

Shatdal, et al [5] improve cache behavior of several widely used algorithms for join and aggregation. The work also contributes a categorization of cache-conscious methods. Methods of the first category try to exploit temporal locality and reuse the data previously loaded in the cache by reducing the working set of the algorithm, while methods of the second category use spatial locality and partition the process in such a way that each part works with a relatively small block of memory that fits in the cache entirely. Kersten et al. [7] employ radix-clustering scheme to attain even better cache performance of join operation.

Our work is based on the idea of join indices proposed by Valduriez in [10]. Given two relations R and S and a join criterion, binary relation of pairs (r, s) (where r and s are RIDs of records of R and S which can be joined using the join criterion) is built. The index on r for this relation is then used to find all RIDs of records of S joined with the given record of R. While on the logical level our multi-indices resemble those proposed by Valduriez, we focus on efficient physical representation of index records under specific circumstances of main-memory database system. At the same time we do not bind the idea with the particular index structure. Like domain indices [6], multi-indices may index more than two relations. We also propose further development of the idea and claim that in many cases significant speed-up can be gained by placing pointer to the index record into the data record and avoiding the index search altogether.

3 Creating and using multi-indices

The main idea of multi-indices is to store information about all records of different relations containing given value of an attribute in the same place. Let $R_1, \ldots, R_k$ is a set of relations with common attribute A. *Multi-index* of the relations $R_1, \ldots, R_k$ on the attribute A is a mapping

$$I^A_{\{R_1,\ldots,R_k\}} : Domain(A_{R_1}, \ldots, A_{R_k}) \rightarrow \{r_i\}.$$

where $Domain(A_{R_1}, ..., A_{R_k})$ is a union of attribute A's domains in relations $R_1, ..., R_k$ and r_i is a record of a relation R_l for some $l = 1..., k$. Notice that the multi-index $I^A_{\{R_1,...,R_k\}}$ can serve as a conventional index for any of the relations $R_1, ..., R_k$ on the attribute A - we only need to filter out records of other relations. We call k the *degree* of the multi-index.

We also define *index record* as a structure containing RIDs of records with the given value of the attribute. The nature of RID depends on the implementation and the storage model being used. We investigate methods for efficient organization of index record later in this work.

3.1 Organization of multi-indices

There can be different ways for organizing multi-index $I^A_{\{R_1,...,R_k\}}$. For simplicity, let us assume we have two relation R_1 and R_2 with the common attribute A. We can then build two indices on A for R_1 and R_2 with shared index records, so that a value of the attribute can be mapped to records of R_1 and R_2 with this value of the attribute by single lookup in either index for R_1 or R_2. As an alternative, we can use single index structure for both R_1 and R_2. Either approach has its advantages, and choice between them should be made taking into account several factors.

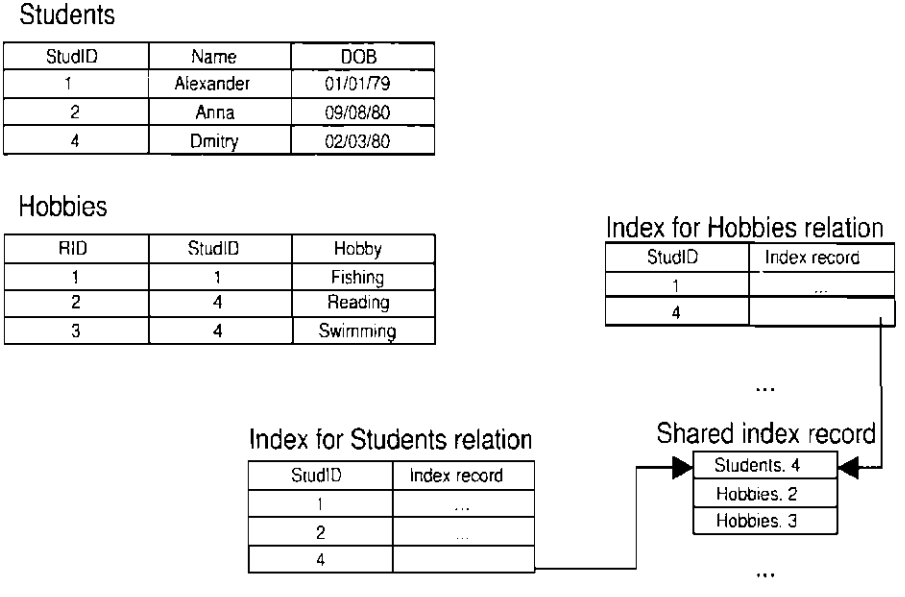


Figure 1. An example of relations {Students. Hobbies} and multi-index on StudId attribute

First, distributions of A's values in relations R_1 and R_2 can differ significantly, so if a single index structure is used for both relations, index lookup is less efficient than it would be if two separate index structures were used. Second, a single index structure for a set of keys occupies less memory than two index structures mapping the same set of keys even if two structures share index records. Besides possible duplication of keys in both structures, there are also auxiliary elements like link pointers etc. Third, as we will show, implementation of multi-indices in the form of several index structures with shared index records degrades performance of insert and delete operations, so for environment with intensive modifications we may prefer a single index structure.

Choice among different variants of implementation of multi-index is therefore a trade-off between storage space, search and update efficiency. If there are three relations with common attribute, the situation becomes even more complicated, as we now have 5 options for constructing multi-index. It is known that the number of all possible partitions of a set (and therefore the number of options for constructing multi-index) is described by Bell numbers [9]. Generally, the issue of choosing optimal implementation of multi-index can be treated as an optimization problem where we are trying to minimize average search cost under restrictions on the amount of main memory available and overhead for insert and update operations. Space limits preclude us from detailed description.

In Figure 1 an example of two relations $\{Students, Hobbies\}$ and a multi-index on the *StudId* attribute is depicted. In the example an index on *StudId* is built for both relations, and indices share common index records. Assuming that only some of students have hobbies, such an organization seems to be better than the single index structure.

We propose the following index record layout. Index record header contains the length of the index record and a table $(RelId \rightarrow offset)_{i=1,\ldots,m}$ that maps unique relation identifier to the offset of the first record identifier from the relation in the index record. Following the record header, record identifiers from the relations $RelId_1$, ..., $RelId_m$ are stored. The main idea is therefore to group record identifiers from the same relation together. Compared with simple sequential layout, this organization incurs some overhead during update of the index record, because we may need to shift part of the record to accommodate to new relation boundaries or even reallocate the record if no free space available inside the record. If, however, modifications to the database are rare as compared with queries, this overhead is well rewarded by the resulting cache efficiency. Fetching record identifiers for relation l costs $1 + \frac{size(I_l)}{size(cache\_line)}$ cache misses in the worst case (where I_l is the size of the block of records identifiers from relation l); fetching records identifiers for relations l and m costs $1 + \frac{size(I_l)+size(I_m)}{size(cache\_line)}$ cache misses.

3.2 Maintaining and using multi-indices

An algorithm for equi-join of two relations R_1, R_2 on common attribute A using multi-index $I^A_{\{R_1,R_2\}}$ is straight-forward. We simply traverse the multi-index

collecting matching records. Notice that an optimization is feasible in the index record layout: if we have a multi-index $I^A_{\{R_1,\ldots,R_k\}}$ and there are expectations that two particular relations R_l and R_m will often be joined in queries, we may place record identifiers from these relations contiguously in the index records thereby decreasing the probability of the extra cache miss. Another optimization is to place record identifiers from the most frequently using relations closer to the beginning of the index record, since this may reduce the number of cache misses occurring during processing of the index record to 1.

We compare the multi-index join with the partitioned hash-join which has been proved to be one of the most efficient equi-join algorithms for MMDBMS [5]. Partitioned hash-join requires preprocessing step that divides relations (say R and S) being joined into partitions based on hash codes of values of the join attribute. This step involves scanning both relations and therefore exhibits poor cache performance because of large number of compulsory cache misses. Each record of the relations should be accessed in order to compute its hash code. The number of cache misses depends on the underlying storage model (one can lay out attributes in such a way that values of the join attribute occupy contiguous region in memory, thereby decreasing number of cache misses required to read them all), but in general for conventional N-ary storage model we may expect one cache miss per each record. After preprocessing step, hash-join builds hash table for each partition of relation R and probes all records in each partition of relation S using this hash-table.

Multi-index does not require any preprocessing, as it already has all the necessary information to perform the equi-join. We only need to traverse the index structure and for each index record form pairs (r,s) of matched records. Notice that conventional index structures like B$^+$-trees and bucket-chained hash tables all support efficient traversal. Of course, we still experience large number of compulsory cache misses. However, we expect this number to be substantially smaller than that during preprocessing step in partitioned hash join, because index records better utilize cache lines and do not pollute them with irrelevant information (such as values of attributes not participating in join). For multi-indices of the smallest degree 2, the cache utilization is especially high, since all the information contained in an index record is required during join. For multi-indices of higher degrees, we have mentioned a couple of optimizations aimed to increase cache utilization. In the multi-index join we also do not need to perform hash code computation, which can be a CPU-bound operation. Given all the above, there are good prerequisites for multi-index join to perform better than the partitioned hash-join.

During insertions, deletions and updates to the database, multi-index, like any other index, should reflect changes in the data. We need to find the index record corresponding to the value of the attribute in the data record being inserted or deleted. If the multi-index consists of the single index structure, we only need to lookup in this structure. If, however, multi-index is a union of several index structures with shared index records we generally need to search in all these indices until the index record is found or the search is exhausted. This may seem

expensive but notice that in practice we almost always know which part of the multi-index should definitely contain the shared index record corresponding the data record being inserted or deleted. For example, when we insert data record into or delete from details relation in 'master / details' scheme, we do know the RID of the corresponding master record, so we only need to search in the part of the multi-index for the master relation. The same applies to the situation when we add or delete fact record from facts relation in the star scheme; in this case we know the RIDs of corresponding records in dimension relations.

Multi-indices can be useful during modifications to the database for enforcing referential integrity constraints [10, 6]. Let us suppose we have three relations: *Students (StudId, Name), Marks (StudId, Subject*, Value) and *Hobbies (StudId, Name)* where *Marks.StudId* and *Hobbies.StudId* are foreign keys for which CASCADE DELETE constraint is enforced. If we need to delete a record from the *Students* relation, multi-index $I^{StudId}_{\{Students,Marks.Hobbies\}}$ enables us to find all records about student's hobbies and marks using single index lookup. In case of two conventional indices on *StudId* for *Marks* and *Hobbies* two lookups would be required.

3.3 Connecting multi-indices and data records

In the above multi-indices were just a generalization of conventional indices; the data records remained unchanged. Unfortunately, such an approach limits usefulness of multi-indices. To demonstrate this, we return to the Students example. Let us suppose we have to find all marks of a student given the student's name: SELECT Subject, Value FROM Students, Marks WHERE Students.Name = 'Anna' AND Marks.StudId = Students.StudId. Good query execution plan first makes use of index *Students.Name* (which we assume to exist), and only then performs the join. The join is therefore used to find all records about student's marks given the *StudId*. But this just as well could be done with conventional index on *StudId* for the *Marks* relation. An alternative execution plan could first join *Students* and *Marks* using multi-indices and then filter out all records which bear nothing to Anna, but this plan looks unreasonable because of low selectivity of the query.

We can make multi-index more useful if we directly connect data records with the corresponding index records as illustrated in Figure 2. If we place a pointer to the index record into data record of the *Students* relation, we will be able to perform the query quite efficiently : we will only need to find a record about Anna in the *Students* relation and then follow the pointer to the index record where we fetch RIDs of corresponding records from Marks relation. This query execution plan does not involve any index search for performing the join. Notice that such a modification of data record's structure is especially simple for MMDBMS, because in MMDBMS we keep relations and multi-index in main-memory, and we do not have to perform any additional address translation when moving indices and relations from disk to memory and vice versa.

Of course, placing a pointer to the index record into the data record has some negative consequences, most obvious of which is that the pointer occupies extra

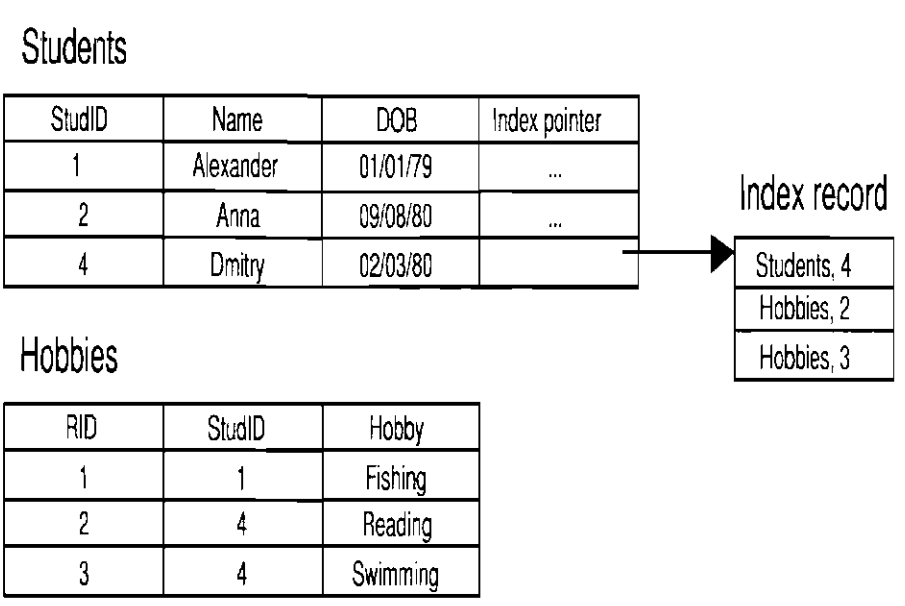


Figure 2. Direct references from records of the *Students* relation to index records

storage. Another unwanted side effect is possible decrease of cache utilization by data records. For those queries which do not include join, the pointer will simply waste valuable cache space. A possible solution to this problem is an adaptive data layout technique like one discussed in [4], which lay out contiguously attributes frequently occurring together in queries. Third issue is that after inserting a new data record the corresponding index record can be reallocated, and we will need to update all pointers to it. However, this can be done efficiently, because all data records holding pointers to the index record are referenced in the index record itself, so we can easily update them having the index records in hand. Generally, we expect that in many cases the proposed modification pays off and significantly increases effectiveness of multi-indices.

4 Implementation and experimental study

We are currently working on implementation and tuning of multi-indices in Memphis, our main-memory research database kernel. The primary goal of Memphis project is to provide a flexible architecture that would allow to support various storage models and index structures, easily plug compression algorithms and include new data types extending the predefined set of built-in types. The salient feature of Memphis is that it is written entirely in a managed language C# with performance-bound places using unsafe code [12]. This greatly facilitates integration of Memphis with applications targeted .NET framework. We

are going to use Memphis as an underlying storage system for the repository containing large amount of information about source code of software projects (this information includes such code objects as classes, methods, properties and relationships among them).

We have implemented some conventional join algorithms used in MMDBMS (partitioned hash join, blocked nested loops, index) as well as multi-index joins. We use bucket hashing for multi-indices; for better efficiency we provide specialized representations of index records for multi-indices of several small degrees. These specialized representations contains fixed number of offset fields in place of general offset table described in section 3.1. We have not implemented references from data records to index records yet.

Our preliminary experiments show that multi-indices work well in important situations, in particular, for joins involving one master and several details relations (which is a very common case in practice). We present some experimental results here. These results only concern performance of join operation; measurements of insert and update overhead caused by multi-indices as well as a more detailed study of joins will be given in the extended version of the paper. We tested our implementation on several datasets, including the real dataset and synthetic ones. Synthetic datasets were generated by a tool that is a part of the Memphis project. This tool takes such properties of the dataset as relations cardinalities, types of attributes, and distribution of attribute values and produces relations in the form of plain text files that can be bulk-loaded into memory of the Memphis system. Each experiment was performed on the cold system; that is, the system was started, all the relations were loaded, then the join was performed. The presented measurements are average values obtaining by running each test several (5-6) times. The experiments were conducted on a conventional workstation (Intel P4 2.8 GHz, 1 Gb RAM). In our tests we compared the performance of the join algorithm using multi-index with in-memory partitioned hash-join algorithm ([5]). The number of partitions in the hash-join algorithm was chosen to provide the optimal performance. Results of experimental runs for some datasets are summarized in the Figure 3.

Dataset	$\|R\|$	$\|S\|$	$T_{HashJoin}$	$T_{CreateMultiIndex}$	$T_{MultiIndexJoin}$	$MultiIndexSize$
Classes	8000	6000	1.6 sec	0.8 sec	1 sec	0.4 Mb
SynthI1	30000	50000	1.2 sec	1 sec	0.85 sec	0.6 Mb
SynthI2	100000	300000	7.6 sec	4.8 sec	7.7 sec	1.8 Mb
SynthS1	100000	300000	43 sec	24 sec	8 sec	2.1 Mb

Figure 3. Results of experimental runs

For each dataset $|R|$ and $|S|$ are cardinalities of the relations being joined, $T_{HashJoin}$ is the running time of the partitioned hash-join algorithm, $T_{CreateMulti}$ is the time required to build the multi-index. $T_{MultiIndexJoin}$ is the running time of the join algorithm using multi-index, and $MultiIndexSize$ is the amount of memory occupied by the multi-index. Since in this experiment we assumed that

the multi-index is built during the join processing, the total running time of the multi-index join is $T_{CreateMultiIndex} + T_{MultiIndexJoin}$. In practice, however, the multi-index is maintained up-to-date, so the first term $T_{CreateMultiIndex}$ disappears from the latter formula.

Dataset *Classes* contains information about classes (basically, names of classes and their members) defined in source code of a large object-oriented system. This information is stored in the normalized form. The relation R ('Classes') contains $ClassID$ and $ClassName$ attributes. It is joined with the relation S ('Members') that consists of $ClassID$ and $MemberName$ attributes, where $S.ClassID$ is a foreign key referring to the $R.ClassID$.

*SynthI*1 and *SynthI*2 are synthetic datasets that include two relations, each of which contains one integer and one string attribute. Integer attributes are unformly distributed in the interval [1, 100000]. The relations are joined using integer attributes as the join attribute. Finally, *SynthS*1 contains two relations, with two string attributes in each relation. The pair of joined attributes are short strings (3-5 characters) that mimic e.g. item code represented in a string form. Note that in case of string attributes the multi-index join provides the noticeable speed-up, as it dramatically reduces the number of expensive string comparisons needed to compute the join result.

5 Conclusion

In this paper we have presented multi-indices - a generalization of conventional indices commonly used in databases. Multi-indices essentially precompute information needed to perform equi-joins by mapping an attribute's value to all the records of different relations containing this value. This enables very efficient processing of equi-joins. We have discussed trade-offs for constructing multi-index, the layout of index records optimized for better cache utilization, maintenance and usage of multi-indices. We have also proposed a further development of the idea - connecting index records with data records, which enables to avoid search in the index structure and leads to even better performance.

We believe that multi-indices can be an effective tool for main-memory database systems. Currently we are working on tuning and experimental study of multi-indices in our research database kernel Memphis. Preliminary experiments show that multi-indices perform well in practically important situations. We will present detailed experimental results in the extended version of this paper.

References

[1] Anastassia Ailamaki, David J. DeWitt, and Mark D. Hill. Data page layouts for relational databases on deep memory hierarchies. *VLDB Journal*, 11:198–215, 2002.

[2] David J. DeWitt, et al. Implementation techniques for main memory database systems. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 1–8, 1984.

[3] H.V. Jagadish, et al. Dali: A high performance main memory storage manager. In *Proceedings of the 20th VLDB Conference*, 1994.

[4] Richard A. Hankins and Jignesh M. Patel. Data Morphing : An Adaptive, Cache-Conscious Storage Technique. In *Proceedings of the 29th VLDB Conference*, 2003.

[5] Ambuj Shatdal, Chander Kant, and Jeffrey F. Naughton. Cache Conscious Algorithms for Relational Query Processing. In *Proceedings of the 20th VLDB Conference*, pages 510–521, 1994.

[6] Changho Kim. Join Processing and Domain Indices, 1991.

[7] P. A. Boncz, S. Manegold, and M. L. Kersten. Database Architecture Optimized for the New Bottleneck: Memory Access. In *Proceedings of the 25th VLDB Conference*, pages 54–65, 1999.

[8] Jun Rao and Kenneth A. Ross. Making B+-Trees Cache-Conscious in Main Memory. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 475–486, 2000.

[9] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics. A Foundation for Computer Science.* Addison-Wesley, second edition, 1998.

[10] Patrick Valduriez. Join Indices. *ACM Transactions on Database Systems*, 12:218–246, 1987.

[11] Jingen Zhou and Kenneth A. Ross. Buffering Accesses to Memory-Resident Index Structures. In *Proceedings of the 29th VLDB Conference*, 2003.

[12] C# Language Specification. ECMA-334 International Standard, 2001.

METAMODELS AND DB TECHNOLOGIES

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
117--126 P.

Using Association Rules to Extract Regularities from Data

Peter Grabusts

Institute of Information Technology
Technical University of Riga
1 Kalku Street, Riga, Latvia
peter@ru.lv

Abstract. Data Mining, the extraction of hidden predictive information from large databases, is a powerful technology with great potential to help users focus on the most important information in their data warehouses. This paper studies classical association rule construction techniques. A method of association rule obtaining initially developed to analyse the consumer basket has turned to be a good tool for other tasks too. The method helps search and find regularities of the form $X \Rightarrow Y$ in different kinds of data. This method is currently widely applied in the tasks of large scale database processing and analysing. The paper presents an algorithm for association rule construction, and contains an implementation example. An analysis of the experiments is given.

Keywords: Data Mining, association rules, Apriori algorithm

1. Data Mining tasks

Over the last decades we have seen an explosive growth in human's capabilities to generate and collect data. Advances in scientific data collection, the widespread introduction of bar codes for almost every commercial product, and the computerisation of many businesses and government transactions have generated a flood of data. It is not realistic to expect that human experts carefully analyse all this data. A significant need exists for a new generation of techniques that can intelligently and automatically transform the processed data into useful information and knowledge. Thus, data mining has become a research area with increasing importance.

Data mining, which is also referred to as knowledge discovery in databases, means a process of nontrivial extraction of implicit, previously unknown, and potentially useful information (such as knowledge rules, constraints, and regularities) from data in databases [1,2]. The general idea of discovering "knowledge" in large amounts of data is both appealing and intuitive, but technically it is difficult.

Data mining is an inter-disciplinary subject formed by the intersection of many different areas. Researchers in knowledge-based systems, artificial intelligence, machine learning, knowledge acquisition, statistics, spatial database, and data visualisation have also shown great interest in data mining.

Many researchers give the following global data mining tasks: classification, clustering, association rules, sequence and prediction.

2. Possibility of association rules analysis

Frequently the needs of business stimulate the development of new methods of intelligent database analysis, which are oriented towards practical business applications. As an example, one of the problems can be mentioned, which shop managers often face: when a customer purchases specific article, then X % of the time he also buys another article that is first article dependent. Say, if he buys bread and butter, then 90% of the time he also buys milk. Initially, that task was used to find a pattern of typical market basket in supermarkets, that is why it is frequently called *market basket analysis*. The regularities, which could evidence for such events' relationships, were called *associations*. Associations or association rules enable one to find relationships among several dependant events. The underlying statement for those rules is the following: if event A has occurred, then event B will also occur with probability X%.

The basics of association rule acquisition are theoretical assumptions about the existence of such rules made by Agrawal, Imielinski, and Swami [3]. In 1994 an effective algorithm for association rule mining was published [4]. The above studies have stimulated the development of numerous similar algorithms, which made it possible to analyse, for example, large scale shopping operations, and extend the task to one of the fundamental methods of intelligent data analysis. Association rules can be employed not only for market basket analysis. They can be applied to any data analysis by carefully examining the regularities found.

Let us consider a short common example that illustrates the essence of association rules. Table 1 contains data on shopping transactions.

Table 1. Market basket

Market basket id	Market basket content
1	juice, water
2.	milk, juice, bread
3.	juice, butter
4.	juice, bread, water
5.	bread

From the data given in Table 1 the following rules can be derived:

- 80 % of all transactions contain juice (1,2,3,4);
- 40% of all transactions contain water (1,4);
- 50% of the transactions containing juice also contain water (1,4);
- all the transactions containing water also contain juice.

The rules derived are simple, understandable and applicable. They are typical for the process of association rule mining. An association rule is an expression $X \Rightarrow Y$, where X and Y are sets of items. The intuitive meaning of such a rule is that

transactions of the database which contain X tend to contain Y. Association rules can be formally defined as follows:

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called *items*. Let $D = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions, where each transaction, t, is a set of items such that $t \subseteq I$. Each transaction is associated with an identifier, called TID. Given an itemset $X \subseteq I$, a transaction t contains X if, and only if, $X \subseteq t$. The itemset X has *support*, s, in the transaction set D if s% of transactions in D contain X; we denote s – support (X). An association rule is an implication of the form X => Y, where X, $Y \subset I$, and $X \cap Y = 0$. Each rule has two measures of value, support, and confidence. The support of the rule X => Y is support $(X \cup Y)$. The *confidence*, c, of the rule X => Y in the transaction set D means c% of transactions in D that contain X also contain Y, which can be written as the ratio support $(X \cup Y)$ / support (X).

Support indicates the frequencies of the occurring patterns, and confidence denotes the strength of implication in the rule. Given a user specified minimum support (called minsup) and minimum confidence (called minconf), the problem of mining association rules is to find all the association rules where support and confidence are larger than the user defined minsup and minconf. It can be decomposed into two subproblems:

1. The large itemsets. Find all itemsets that have support above the predetermined minimum support. These itemsets are called large itemsets. Sometimes they are also called frequent itemsets.
2. For each large itemset, derive all rules that have more than the predetermined minimum confidence as follows: for a large itemset X and Y, where $X, Y \subset I$, and $X \cap Y = 0$, if support $(X \cup Y)$ / support $(X) \geq$ minimum-confidence, then the rule X => Y is derived.

The overall performance of mining association rules is determined by the first step. After the large itemsets are identified, the corresponding association rules can be derived in a straightforward manner.

Let us illustrate the aforementioned with an example. Table 2 represents transaction database. It is a set of products I that consists of products A, B, C, and D.

Table2. Market TID

TID	Market basket	TID	Market basket
1	{A,C}	6	{A,B}
2	{B}	7	{A,D}
3	{A,B,C,D}	8	{B,C,D}
4	{B,D}	9	{C,D}
5	{A,B,D}	10	{A,B,D}

Each row of the table contains transaction identifier TID, which characterises the number of customer's performed operation as well as a set of the goods purchased. The support of itemset {A,B} is 0.4. The value of support of itemset {A, B, D} is 0.3. Hence, the confidence of rule $\{A,D\} \Rightarrow \{B\}$ is 0.75. From this it follows that

$$c(A, D \Rightarrow B) = \frac{s(A \cup B \cup D)}{s(B)} = \frac{0.3}{0.4} = 0.75$$

If the boundary value of support, *minsup*, is less or equal to 0.3 but the boundary value of confidence, *minconf*, is less or equal to 0.75, the rule is being considered acceptable. It can be concluded that this association rule is derived: "When a customer buys products A and D, it is possible that in 75% of cases he will buy product B as well".

The statements of that kind are without doubt acceptable only for large databases. Support and confidence values do not yet guarantee the suitability of the rule for modelling customer's behaviour. They can only assist in making decisions.

3. The Apriori algorithm for rule mining

Various algorithms have been proposed to discover the large itemsets, see, for example, [3] and [4]. The Apriori algorithm is one of the most popular algorithms in the mining of association rules in a database.

In mining association rules, the problem of finding the large itemsets is fitting the above general description. The goal is to find the large k-itemsets; this problem can be solved if the large (k-1)-itemsets were found. One can use large (k-1)-itemsets to generate candidate k-itemset. The optimal policy is that candidate k-itemset's support is greater than the user defined support. Solve the problem of finding large (k-1)-itemsets relying on the solution of large (k-2)-itemsets, and so on.

(1) The Apriori algorithm is shown in Figure 1 and Figure 2. The large itemsets are computed through iterations. In each iteration the database is scanned one time, and all large itemsets of the same size are computed. In the first iteration, the size-1 large itemsets are computed by scanning the database once. Subsequently, in the k-th iteration ($K > 1$), a set of candidate sets, C_k, is created by applying the candidate set generating function AprioriGen on L_{k-1}, where L_{k-1} is the set of all large (k-1)-itemsets found in iteration k-1. AprioriGen generates only those k-itemsets whose every (k-1)-itemset subset is in L_{k-1}. The support counts of the candidate itemsets in C_k are then computed by scanning the database once, and the size-k large itemsets are extracted from the candidates.

```
Input: A database D of transactions and support threshold minsup
Output: A set L that contains all frequent itemsets of D
L1 = {large 1-itemsets}
for (k = 2; Lk-1 ≠ 0; k++ ) do begin
    Ck = AprioriGen (Lk-1)      // new candidates
    for all transactions t ∈ D do begin
       Ct = subset (Ck, t)      // candidates contained in t
      for all candidates c ∈ Ct do
          c.count ++
    end
    Lk = {c ∈ Ck | c.count ≥ minsup}
end
answer = ∪k Lk
```

Figure 1. The Apriori algorithm.

Apriori candidate generation. The AprioriGen function takes as an argument L_{k-1}, the set of all large (k-1)-itemsets. It returns a superset of the set of all large k-itemsets. First, in the join phase, L_{k-1} is joined with itself, the join condition being that the lexicographically ordered first k-2 items are the same, and that the attributes of the last two items are different. Second, in the subset pruning phase, all itemsets from the join result which have some (k-1)-subset that is not in L_{k-1} are deleted.

```
function AprioriGen(L_{k-1});
// the join phase
 insert into C_k
  select p.item_1, p.item_2,...., p.item_{k-1}, q.item_{k-1}
  from L_{k-1} p, L_{k-1} q
  where p.item_1=q.item_1,....p.item_{k-2}=q.item_{k-2}, p.item_{k-1}<q.item_{k-1};
// the prune phase
 forall itemsets c ∈ C_k do
   forall (k-1)-subsets s of c do
     if ( s ∉ L_{k-1}) then
       delete c from C_k;
```

Figure 2. Function ApioriGen.

(2) Generating rules. For every large itemset l, we output all rules a ⇒ (l-a), where a is a subset of l, such that the ratio support (l) / support (a) is at least *minconf*. For example, if the rule A ∧ B ⇒ C ∧ D holds, then the rules A ∧ B ∧ C ⇒ D and A ∧ B ∧ D ⇒ C must also hold. The rule generation algorithm is described in Figures 3 and 4.

```
For all large k-itemsets l_k, k ≥ 2, do begin
    H_1 = {consequents of rules from l_k with one item in the consequent};
    ApGenrules (l_k, H_1);
End
```

Figure 3. Rule generation algorithm.

```
Procedure ApGenrules (l_k: large k-itemset, H_m: set of m-item consequents)
   If (k > m+1) then begin
      H_{m+1} = AprioriGen (H_m);
      For all h_{m+1} ∈ H_{m+1} do begin
         Conf = support (l_k) / support (l_k - h_{m+1});
         If (conf ≥ minconf) then
            Output the rule (l_k - h_{m+1}) ⇒ h_{m+1}
                  with confidence = conf and support = support (l_k);
         Else
            Delete h_{m+1} from H_{m+1};
      End
      ApGenrules (l_k, H_{m+1});
   End
```

Figure 4. Function ApGenrules.

From a large itemset l, the algorithm first generates all rules with one item in the consequent. The algorithm then uses the consequents of these rules to generate all possible consequents with two items that can appear in a rule generated from l, etc.

4. Experiments

The main motivation for experiments was the desire to find regularities in raw data using association rules mining method. The author did not have an opportunity to use professional software packages such as the data mining tool Clementine or others therefore the experimental part was carried out in the Matlab environment.

Let us now consider the application of association rule mining algorithm to statistical data analysis, so as to find data characterising values and obtain possible regularities. At the first stage, data for the experiment were prepared. In the beginning, all the data were in the SPSS format and during that stage all the parameters were fixed and a system of codifiers was elaborated so as to make it possible to apply the association rule mining algorithm to the data. At the second stage, the association rule mining algorithm was implemented in the program form and the data characterising rules at the initial boundary restrictions were obtained.

A series of experiments were then performed aimed to ascertain the dependence of the count of obtained rules on the support's initial values. As experimental data, the data of the Latvian Central Statistics Office about reply variants obtained from 3044 respondents. The data selected for the experiments were related to the study of inhabitant migration process. The respondents were asked the following questions:

1. In which country were you born? (with 11 possible reply options offered);
2. How long have you lived in this place? (with 4 reply options offered);
3. Where did you live before removing to this place? (with 3 reply options provided);
4. Please designate the type of the place you lived in before removing to the current place? (with 7 possible reply variants);
5. What was the reason for you to move to the current place? (with 6 possible replies);
6. Are you planning to move to another place within the next 3 years? (with 5 possible reply options provided).

The goal of the experiment was to determine possible relationships in these data and to determine the dependence of the count of regularities on the preliminarily assigned boundary values of support and confidence.

In the first part of the experiment it was assumed that the confidence was minconf=95 and the support was minsup=95. See Figures 5 and 6 for the obtained graphs of rule support analysis and rule confidence analysis. At these initial values as many as 58 rules were derived. For each rule there were calculated support value and confidence value.

Below one can find some derived rules with greatest support values:

12 74 => 51 Support=592 and confidence is 98;

46 => 83 Support is 545 and confidence is 97;

12 74 83 => 51 Support is 530 and confidence is 98;

12 46 => 83 Support is 443 and confidence is 97;

45 51 65 => 83 Support is 303 and confidence is 95.

Taking into account the decoding results one can conclude that:

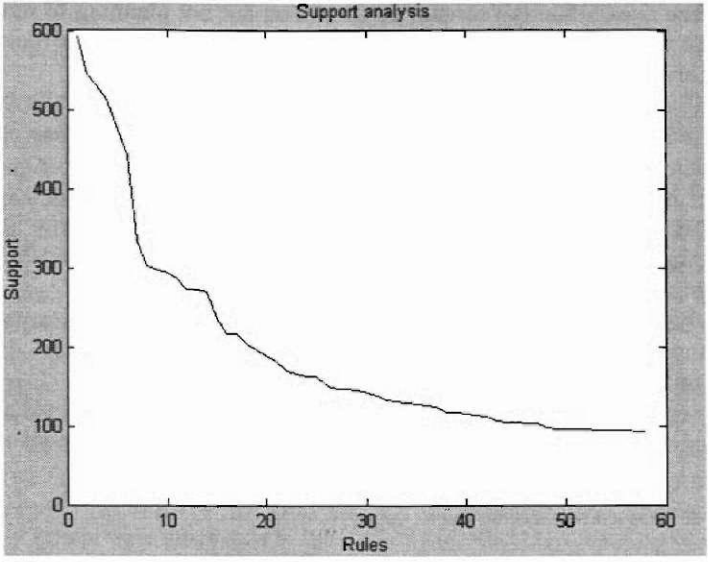


Figure 5. Dependence of rule count on support values.

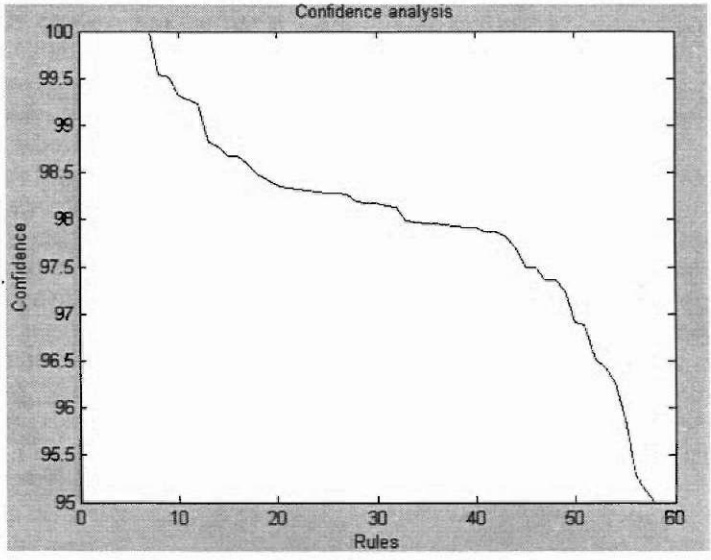


Figure 6. Dependence of rule count of confidence values.

- the 1st rule determines: IF "you were born in Latvia" AND "moved to the current place due to family reasons" THEN "Before moving to the current place you were living in Latvia".

- the 2nd rule determines: IF "You have always lived in this place" THEN "In the next 3 years you are not planning to move to any other place".
- the 3rd rule determines: IF "You were born in Latvia" AND "You have moved to this place due to family reasons" AND "You are not planning to move to any other place within the next 3 years" THEN "Before moving to the current place you lived in Latvia"
- the 4th rule determines that: IF "You were born in Latvia" AND "You have always lived in this place" THEN "You are not planning to move to any other living place within the next 3 years"
- the 5th rule determines that: IF "You have lived in this place up to the age of 50 years" AND "Before moving to this place you lived in Latvia" AND "Before that you lived in a village" THEN "You are not planning to move to any other living place within the next 3 years".

By analysing the rules derived one can state that the persons interrogated live in a small compact living place with a small migration trend. The rules derived are logically understandable and represent a real-world situation.

In the second part of the experiment, rule count values were obtained at different support values and fixed boundary confidence values. The results are given in Table 3.

Table 3. Rule count's dependence on support values

Support	10	15	20	25	50	75	100
Conf.-50	2690	2071	1628	1346	765	509	386
Conf.-75	1717	1343	1029	839	473	318	241
Conf.-90	983	736	553	436	246	159	121

The graphic form of the above correspondence can be seen in Figure 7.

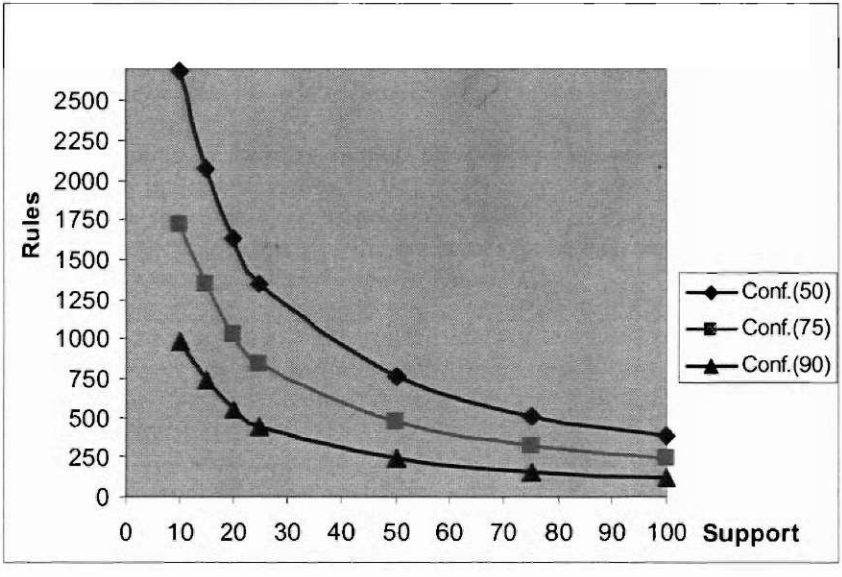


Figure 7. Graph of rule count's dependence on support values.

Based on the table data and graphically given correspondence, one can conclude that the greater the assigned confidence level and support's boundary value is, the smaller the number of association rules discovered is and hence the rules are stronger.

5. Conclusions

This paper examined association rule mining techniques. The mechanism of association rule mining, which was initially intended for the market basket analysis, turned out to be a good tool for a wider range of tasks. With the help of that mechanism, it is possible to mine and discover regularities of the form $X \Rightarrow Y$ in different data types. Nowadays it has recognised widespread application in tasks of large database processing and analysis. Association rule mining method justly lies among the main intelligent data processing methods [5,6].

Association analysis can be useful as one of the first study's steps when there is only known (or essential) one of the data characterising parameters. The main advantage of association rules as compared to other cause-effect discovering methods is a fairly simple rule generation. The rules discovered are easy to perceive and to interpret. The rules of that kind are easy to understand and, respectively, can be used directly in data analysis. On the other hand, the rules generated are actually operators of several programming languages, (for example, SQL) but the method can be easily associated with databases. Another advantage of association rules is a possibility of working with records of different length. Finally, the method can be conveniently employed at the initial stage of data analysis when there is no clear understanding about the data being analysed and it is not clear how to deal the particular task.

At the same time it should be noted that association rule analysis has its own bottlenecks whose study could turn to be a valuable application area:

- Software implementation of association rules requires considerable time;
- The analysed data should be possibly homogeneous.
- Unfortunately, erroneous or strange data also participate in rule formation.

Association technique best performs in the cases when various parameters are fairly frequent in the datasets. Otherwise, the rules will only connect frequently repeated parameters and we will learn nothing new about rarely met parameters, that is, time will be non-effectively spent for processing less important rules.

It can be concluded that the mechanism of association rule mining is very effective for certain classes of task. It is, however, important to recognise that the association rules discovered require a thorough analysis to make their application effective.

The main achievement for the author while doing this research was his acquired belief that the association rules method is a good instrument for finding regularities. In the future research will be carried out to compare the achieved results with other data mining methods.

Acknowledgements

I thank Professor Arkady Borisov (Riga Technical University) for useful comments on the paper.

References

[1] H. P. Newquist. Data Mining: The AI Metamorphosis. Database Programming and Design. № 9, 1996.

[2] U. M. Fayyad, G. Piatesky-Shapiro, P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In U. M. Fayyad, G. Piatesky-Shapiro, P. Smyth and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, chapter 1, AAAI/MIT Press, Menlo Park, California, USA, 1996.

[3] R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules Between Sets of Items in Large Databases. Proc. Conf. on Management of Data. ACM Press, 1993.

[4] R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules. Proceedings of the 20th International Conference on Very Large Databases, 1994.

[5] M. Klemettinen, H. Mannila, P. Ronkainen, T. Toivonen. Finding Interesting Rules from Large Sets of Discovered Association Rules. In 3rd International Conference on Information and Knowledge Management (CIKM), November, 1994.

[6] M-S. Chen, J. Han, P. S. Yu. Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, December 1996.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673. Databases and Information Systems
127 138 P.

Improvement of the dataware system for decisions making at the railway

Vasilijs Demidovs

Doctoral (PhD) student, Transport and Telecommunication Institute
Lomonosova1, Riga, LV1019, Latvia.
Ph: (+371)-5834067. Fax: (+371)-5834366. E-mail: dem@ldz.lv

1. General description of the research work

The present paper offers for consideration the results of investigating the problems of improving the Decision Support Systems (DSS) at railway transport with the account of the Latvian railway information systems development and latest technologies in the sphere of databases, which results have been gained by the author in the period from 2001 to 2004. Supervisor of doctoral studies – professor Dr. habil. sc. ing. Jevgenijs Kopitovs.

Actuality of the problem. Modern railway has some specific peculiarities, which affect greatly the efficiency of the decisions to be made. We should mention the following:

- high dynamics of processes;
- random factors;
- high reliability and safety;
- large financial, labour and material resources;
- large distances between related objects;
- complex map of Railway.

Errors in planning of transportation processes result in feasible decrease of the efficiency of the enterprise activity and of the clients' service quality. Therefore, searching of optimal decisions for the forthcoming periods of functioning is quite a complicated job.

At present Latvian railway (State Joint-Stock Company «LDz») is working out and deploying Information Systems (IS) responsible for different spheres of its business. The data generated by these systems enable to evaluate the efficiency of the performance and exercise prospective planning. To resolve these tasks each system has developed individually specialized local programming complexes meeting the requirements of managers at different levels. But all these programs have been developed on the basis of outdated technologies, they are difficult to support and can, therefore, work only with small volumes of data.

Effective complex planning of the whole enterprise activity is hampered due to the following problems.

- *Non-homogeneity of the employed information support.* For example, the information system of the Latvian railway consists of 5 major IS, such as cargo transport system, passenger transport system, finance system, infrastructure information system and rolling-stock system. Every IS includes a variety of non-

uniform sub-systems, often absolutely unrelated. This is caused by the fact that the sub-systems have been developed separately and on the basis of different technologies and different software and hardware.

• *Huge data volumes.* These systems generate more than 10 000 000 transactions per year.

• *Complicacy of simultaneous application of the same data of IS for resolving the tasks of decision-making.* In the process of the analysis some data are substituted for the hypothetical ones in conformity with possible managing affects. After the analysis the system restores its initial state. During this period other users should be denied access to the data to avoid getting non-correct results.

• *Decrease of the information reliability.* A big variety of local systems generating data, which are difficult to verify, reduce the reliability of the information necessary for making decisions. Often different managers have different information about one and the same object since different programs have been used to get it, and the extent of multitude of the sources of information or of the processed data has been different.

The above statements speak in favor of the actuality of the research aimed at improving IS at railway. And it has become ground for the present research work.

Aim and tasks of the research. The aim of the doctoral research is increasing the efficiency of the DSS at railway by means improving the structure and the process of the information system support functioning.

According to the given aim, the author undertakes to resolve the following main tasks:

1. Analyzing the IS working at railway as well as the particulars of their application in DSS and defining the problems to be resolved.
2. Formalizing the tasks of decision-making and working out the models of the railway transport system functioning.
3. Working out the methodology of building and developing the IS at railway on the basis of the advanced achievements in the sphere of databases.
4. Solving of the application problems connected with the analysis of the railway functioning in the previous periods, forecasting the further behaviour of the IS and evaluating the reliability of the received forecasts and recommendations.

Methodology and methods of research. The dissertation research is based on:

• the results of processing the statistical reports about the activity of railway transporters provided by the company "Pasažieru vilciens";

• statistical data received by the author by means of processing the transactional data about the passengers transported by railway;

• the results of modeling the processes at railway transport;

• the results of the scientific research performed by the Institute of Transport and Telecommunication and LDZ, in which the author had played his personal part;

• the materials of scientific conferences in which the author had directly participated;

• technical documentation, scientific-technical literature and periodic releases dedicated to the problems dealt with by the author of the dissertation.

The methods of research are based on modern theory of system analysis, decision-making theory, theory of sets and theory of probabilities and mathematical statistics.

Scientific innovation. The given work presents a new approach to developing DSS at railway transport making use of modern databases technologies.

In the process of the research there have been received the following major results:

1. There has been developed Conceptual Active Model of Prediction System at railway.
2. There has been offered and tested a method of hierarchically related views, which enables to get a universal mechanism of managing the processes of data transforming and provides a high reliability level of the information received on the basis of the data processing results.
3. There has been offered a method of building virtual models, which application together with clear system formalization allows to increase the level of system access, reduce the time of the data analysis and of the new reports development as well as to provide a guaranteed reliability of the received results.
4. Complex research of forecasting railway transport indicators has been performed.
5. A temporal model for railway IS developed.

Practical value and realization. The undertaken research gave possibility to improve the existing and speed up new analytical and transactional IS at LDz in the Relational DBMS IBM UDB DB2.

The models and methods developed by the author have been used in a number of projects creating the following IS:

- System of analysing and releasing statistical reports on passenger transport;
- Train timetable system;
- Information-analytical system of Information control system of freight traffic.

The results of the research are widely used in educational process of Transport and Telecommunication Institute as sections of the course "Modern Database Technologies" for Master program in Computer Science.

Approbation of the work. The main results of the research have been presented at 10 scientific conferences in Latvia, Lithuania, Russia and Belgium.

Publications. The dissertation is submitted by set of published scientific work. Total number of the publications makes 13, including 7 papers and 6 report theses. They cover the issues of DSS development and of forecasting and searching optimal management decisions for successful activity of transport enterprises. Special attention is given to the methods of processing huge volumes of data having temporal characteristics and to the development of virtual forecasting models.

2. Description of the main research tasks

2.1. Formalization of the decision-making tasks at railway transport

Railway is a very complicated system aggregating all processes and events taking part there and, therefore, developing of DSS for railway could not be called a trivial task. LDz has its own sophisticated scheme of the railway network with a lot of alternative option routes between stations. Let's consider for example one of the

socially important spheres of the railway activity – passenger transport. To analyse passenger streams at LDz there has been developed a model of splitting the railway line into separate sections according to the requirements of the statistical report CO-25. For detailed analysis every line is divided into sections shown in Fig.1, which in turn consist of stations sets [8].

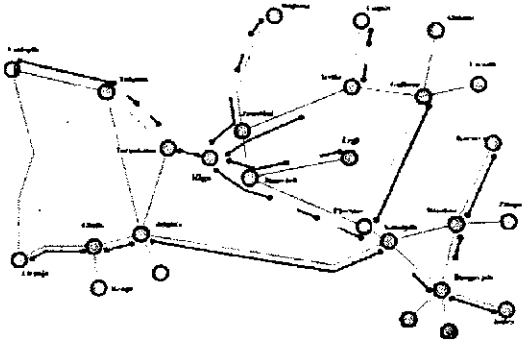


Figure 1. Model of sections CO-25.

These models of lines and sectors are not constant and are changing with time. Several models can exist concurrently. For example, together with the model CO-25 there exists a model developed on the principle of sectors belonging to this or that administration territory. Thus only for one task the system should support several models of the railway network and be able to transform the models in the analysis process.

Calculating passenger streams it is necessary to define the exact route of each trip and to apply it on the corresponding lines and sections. The task is hampered by the fact that the model CO-25 has got overlapped lines and sections and yields some uncertainties, which the system has to process.

To eliminate these uncertainties in the system we should describe a set of rules and restrictions to be considered in calculations. Taking into account the non-permanent character of these rules and restrictions, we should separate them from the models of the railway network, form the mathematical model and define them as those affecting the system.

To formalize task setting and to describe models we'll introduce a group of sets characterizing the railway objects and the system as a whole. As an example some of them are shown below:

- $S = \{s_1, s_2, ..., s_k\}$ – set of all Latvian railway stations, where k – power of set S.
- $P = \{p\}$ – set of couples of stations to trip between, where $\forall p = \{s_i, s_j\}$, $s_i \in S$ and $s_j \in S$, $i \neq j$ and $i, j \in \{1, ..., k\}$.
- $R = \{r_m\}$ – set of all optional trip routes between two couples of stations, where $r_m = \{s_1^{(m)}, s_2^{(m)}, ..., s_\gamma^{(m)}\}$, $s_i^{(m)} \in S$, $i = \overline{1,\gamma}$, $\gamma \leq k$, $m = \overline{1,\beta}$ – route identifier. β – power of set R.
- $L = \{l_q\}$ – set of lines, where $l_q = \{s_1^{(q)}, s_2^{(q)}, ..., s_\lambda^{(q)}\}$, $s_i^{(q)} \in S$, $i = \overline{1,\lambda}$, $\lambda \leq k$, $q = \overline{1,\chi}$ – line identifier, χ – power of set L.

• $U = \{u_z\}$ – set of sectors, where $u_z = \{s_i^{(z)}, s_j^{(z)}\}$, $s_i^{(z)} \in S$ and $s_j^{(z)} \in S$, $i \neq j$, z – sector identifier, and $u_z \subseteq l_q$, which means that both stations of the sector should belong to one line $s_i^{(z)} \in l_q$ and $s_j^{(z)} \in l_q$, where $l_q \in L$.

• $Y = \{y\}$ – set of managing affects (rules and restrictions applied on models of lines and sectors in calculating railway processes).

Using the sets introduced above we can describe all models, rules and restrictions as shown below:

• if the condition $r_i \cap l_j = \varnothing$ is held, the line and the route do not overlap;

• if the condition $r_i \cap l_j = l_j$ is held, the line is completely included into the route;

• if the condition $r_i \cap l_j = r_i$ is held, the route is in the frame of one line;

• if $(r_i \cap l_j \neq \varnothing) \wedge (r_i \cap l_j \neq l_j) \wedge (r_i \cap l_j \neq r_i)$, the line is partially included into the route;

• if $l_i \cap l_j = \{s_\tau\}$, where $\tau = \overline{1,\varphi}$ and $\varphi > 1$, lines l_i and l_j partially overlap.

The given approach is used by the author in building models of the system for resolving tasks of working out the policy of railway activities such as tasks of forecasting, defining optimal tariffs, optimization of drivers and conductors work, introduction of new trains, etc.

To resolve the task of forecasting the system behaviour in the forthcoming periods there have been considered several methods presented below.

• ***Formal extrapolation method.*** In accordance to the given method system status forecast $X^*(t_e)$ for the future moment of time t_e is defined as

$$X^*(t_e) = F\{X(t_i), i = \overline{1,n}\}, \quad (1)$$

where F is a certain functional being a system model per se, $X(t)=\{x_1(t),x_2(t),...,x_k(t)\}$ – vector, which characterizes the status of system at the moment of time t, where $x_j(t)$, ($j = \overline{1,k}$) is the system's indicator observed at the moment of time t.

• ***Active prediction method***, where possible control impacts on the system during the period of forecasting are taken into consideration. In this case, the prediction model of the system status could be written in the following form:

$$X^*(t_e) = F_1[X(t_i), i = \overline{1,n}; Y(t_e)], \quad (2)$$

where F_1 is a certain functional; $Y(t_e)=\{y_1(t_e),y_2(t_e),...,y_q(t_e)\}$ is vector of control impacts on the system $y_i(t_e)$ in the period of time from t_n to t_e.

The problem of prediction of *optimal control decisions* for the future period can be formulated mathematically as:

from set of possible strategies Y_Σ – set of vectors of control impacts on the system in the period of time from moment t_n to moment t_e – to choose vector $Y_{opt}(t_e) \in Y_\Sigma$, where given integral criteria of system efficiency $E[X(t)]$ has its extreme value, i.e.

$$E\left[X^*(t_e)\right] = E\left\{F_1\left[X(t_i), i=\overline{1,n}; Y_{opt}(t_e)\right]\right\} \rightarrow \underset{Y_{opt}(t_e)\in Y_\Sigma}{extr}, \qquad (3)$$

where E is a certain functional, characterizing the efficiency of system operation; ***extr*** is the minimum or maximum.

As it has been already mentioned, extrapolation $X(t)$ in time shows how the events will develop, if all tendencies from the past are saved in future. But if known (accounted) or unknown (unaccounted) factor or group of factors will be changed, then the system will behave in a different way. The factor itself could be unknown to us at present, but we can predict the results of its impact on the observed indices. For example, we cannot know the reasons of dollar rate change to national currency, but we can model extreme situations and see how the model will demonstrate itself, because the consequences of this secondary factor are possible to be detected with a certain degree of probability. Thus, some bifurcations of prediction peculiar to incursive (anticipatory) systems appear [1].

With this aim we can determine vector of random factors, which characterize new impacts of environment on the system. As a result we can obtain the prediction model in the following form:

$$X^*(t_e) = F_1\left[X(t_i), i=\overline{1,n}; Y(t_e); Z^*(t_e)\right] \quad , \qquad (4)$$

where $Z^*(t_e)$ – vector of forecast of random factors, characterizing the impacts of environment on the investigated system in the moment of future time t_e.

Note that the suggested approach can be applied for forecasting railway transportations in accordance with the tariff policy changing as a managing influence on the system with the account of the outside environment influence. Such outside affects taken into account in the process of forecasting, as for example in the case with cargo transportations, can be consideration of other transporters activities inland and abroad, the dollar exchange rate, prices of the transported cargoes, etc.

2.2. Investigating the problems of employing Data Warehouse in railway IS and the ways of their solving

It is suggested to build the above system DSS on the basis of Data Warehouse (DW) where big volumes of data from different sources are accumulated. DW, as compared with Database (DB) of On-Line Transaction Processing (OLTP) systems, most completely describe the system functioning as a whole because they contain historically related data of life activity of several OLTP systems during the whole period of their existence as well as data from outside sources.

Recently DSS has seized to be the prerogative of a certain managing layer and it looks like a pyramidal model and, therefore, it has become subject to stricter requirements in the terms of access data reliability. To make decisions on the basis of historically accumulated data in DW, a physical model of a really existing situation is built.

This model can be also used to forecast the behaviour of the system in the future. For this we should incorporate into the real model those components, which describe

new managing affects and influences of the outside environment, after that the received model is projected on the forecast period. Thus we can get a model of probable system behaviour in the future.

A special part of the forecasting task resolution is the necessity of considering several managing strategies, which could be used in the forthcoming period for different variants of the outside environment behaviour. As a result several model variants are built in which different combinations of managing and outside influences on the system are considered. Application of DW technologies allows building a lot of System Models on the basis of a huge accumulated quantity of data. But the necessity of building a great number of models, in its turn, causes some problems connected with a sharp increase of the volumes of the data stored and with the time necessary for solving the forecasting task. High dynamics of the system leads to the fact that the processes cannot be formalized completely, that's why the procedure of decisions working out is put off for a longer time. And the requirement to the system access comes into contradiction with the requirements to the data reliability as a result of the data being blocked in the process of analysis [5].

The degree of the forecast reliability depends on the degree of the DW data. The DW architecture is based on the schemes of star and snowflake types. These schemes are characteristic of having the table of facts in which all transactions and aggregate transactions are described as well as the tables of measurements for every entity. Here the notion of transaction may differ from the analogical notion in the initial data received from OLTP systems. One of the obligatory measurements of the given scheme is measurement of time. Thus we have a temporal indicator in the data prepared for the analysis. Data formation in a certain period is made on the basis of the Referenced Data System (RDS). The RDS data are subject to constant changes, for example old trains are cancelled and new trains assigned, timetable and train traffic routes are changed. Preparation of the data in a long period requires consideration of all changes in RDS in the processed period. To create a new physical model of data, which is impossible with the existing data, we'll need all the source data. That's why to keep the system in DW flexible, it is necessary to store all the data received from the source systems and make them uniform. Therefore they should also consider the temporal factor reflected in the temporal model [1, 3].

So, in spite of the advantages and the necessity of using DW technologies there is a number of problems to be solved:

- blocking the stored data;
- providing reliability and completeness of the data used;
- temporal character of the data under research;
- big size of relations;
- long delays in processing complicated queries;

The ways of solving the above problems are given below.

2.3.Creating a temporal system as a means of increasing the reliability and completeness of the data in IS

The paper [1] presents the analysis of the peculiarities of temporal principles in the framework of relational model and open model with Abstract Object Identify (AOID) is offered (shown in Fig.2), which provides the support of the multiversion

of the object, and minimizing expenses for "imitation" of DELETE and UPDAT operations with the transition of the old version to Shadow Area. In this mod lifespan of the object is described through lifespans of all its properties, defined i different relations and having time attributes DATE_START (time of lifespan star and DATE_STOP (time of termination).

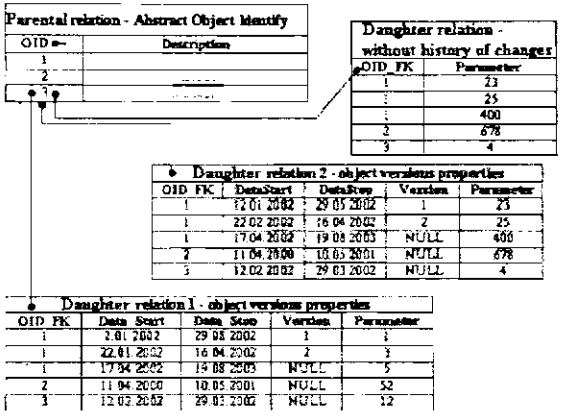


Figure 2. Data Model of Referenced Data System with AOID.

The first parental relation describes unchangeable AOID – one for all its version Properties of the object are described in separate dependent relations, which, as rule, are not parental not for any relation. For one parental relation with AOID the can be more than one daughter relation describing the properties of the object. The characteristics can be divided according to different relations due to the frequency their changing. Seldom changed properties are better to be stored separately fro frequently changed properties.

Such approach lightly complicates the data structure, but it makes the system be open to the changes and reduces the requirements to the disc space for the storag of the multitude of the object versions.

The issues of realization of systems with lifespans periods overlapping a analyzed in papers [3]. Realization of the temporal logics using some elements active databases in analytical applications and real time tasks has been performed o the basis of triggers and Java Stored Procedures. The efficiency of the suggeste method of building temporal BD is illustrated in a sample task with multilev overlapping of objects lifespans in the database of a train timetable shown in Fig.3.

The peculiarity of such approach is in the fact that the state of the object durin the changing can save its actuality, as in past, and in future – only for some period time it is substituted by another versions. All periods of time, which are not include in the category of actual ones are considered to be not only old data, but als "wrong" ones and never existed data. At one and the same time there can exist mo than one tuple describing different properties of one and the same object that h been unallowable in the first example.

Let t_n mean time of observation. Train schedule starts with t_1 to t_6. Later o change of schedule is entered to time period t_2 to t_4, and even later to time period to t_5 one more change is entered. To the query about the train route at the mome

t_{n1}, the schedule t_1–t_6 will be obtained, and at the moment t_{n2} – schedule t_3–t_5, and at the moment t_{n3} – again schedule t_1–t_6.

To provide results reliability working with a complicated objects time structure author has suggested a method of time environments [7]. The user placed in this Temporal Environment has his own view of all objects seen at the moment of time $t=URT_p$ and determined in the relation $R^{(t)}(A_1,...,A_m)$ by the following formula: $R^{(t)} = \sigma_F(R_1) \triangleright\triangleleft \sigma_F(R_2) \triangleright\triangleleft ... \triangleright\triangleleft \sigma_F(R_p)$, where $F: t_s \le t \wedge (t_e > t \vee R_i.t_e \; is \; NULL)$; $\triangleright\triangleleft$ — the operation of the natural join of relations in the common attribute *AOID*. And he does not have to care about the time framework in which this or that object is situated.

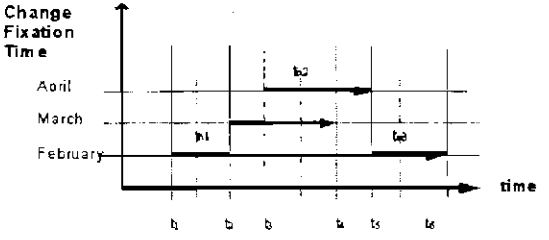


Figure 3. Crossing of Object Lifespans in DB of the Train Timetable System.

If the user does not clearly set the URT_p, then, by default, the URT_p equals the current time $t_{current}$. In this case the variable always changes and corresponds to the special register CURRENT TIME. If the user clearly sets the URT_p, he gets into the "frozen" world at the given moment of time. All queries of the user will be transformed immediately as shown in Fig.4, and will be performed as if a time variable was given in every query for every relation.

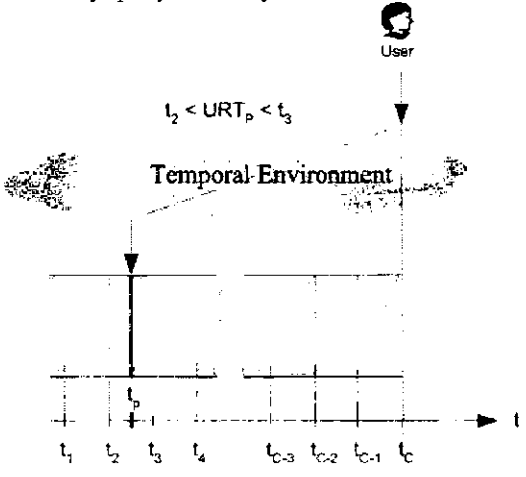


Figure 4. Execution of a query given at a moment t_p.

Having defined the time variable of the environment $URT_p=t_p$ where $t_2<t_p<t_3$, the user can see the objects of the database in the state in which they were at the moment of time t_p.

2.4. Development of Virtual Models and decomposition of information queries with the help of views

Developing virtual models of data allows to reduce the DW size level since it has additional meta-data and Models Repository describing a multitude of virtual models base on a real model and vectors of managing impacts as shown in Fig.5. There is no need of blocking the Real Model data since each analyst works with its own set of Virtual Models.

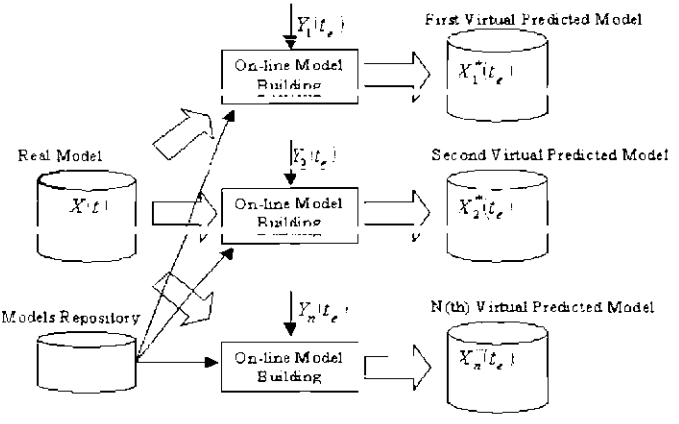


Figure 5. Conceptual schema of Virtual Models building approach.

In papers [4, 5, 6] we have described methods of building virtual models in DSS, which allow to evaluate the current situation, make forecasts for the future and define the required correcting influences to compensate negative tendencies.

Realization of sophisticated algorithms of data transformation in DSS is connected with the problems of providing the received results reliability. To resolve these tasks at the stage of data transformation, it is suggested to use the method of hierarchically related views [2] allowing to get a universal mechanism of managing the data transformation processes and giving additional possibilities in working out and testing the processes. The idea of the given method is to split complicated query logics into separate steps – views with a possibility to test each stage. Thus, transforming one major view into several simple ones we get a graph of interrelated views shown in the right part of Fig.6.

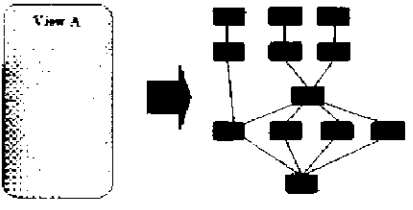


Figure 6. Scheme of transforming a complicated view.

The articles [4, 8] deal with the questions of virtual models application in DSS on the Latvian railway. Let's consider one of examples – the statistical reporting on passenger transportations shown in Fig.7.

The developed conceptual system model consists of five fragments:

- Real Data Model – processed and described data of OLTP systems;
- Models Repository – models describing lines and sectors of a railway and the process of calculating their loads;
- Repository of Control Action – rules and restrictions managing the calculation process;
- Virtual Data Models – virtual data models;
- Description of the process of the models interaction at the stage of data transforming from Real Data Model in Virtual Data Models – $F_j^{(t)}$.

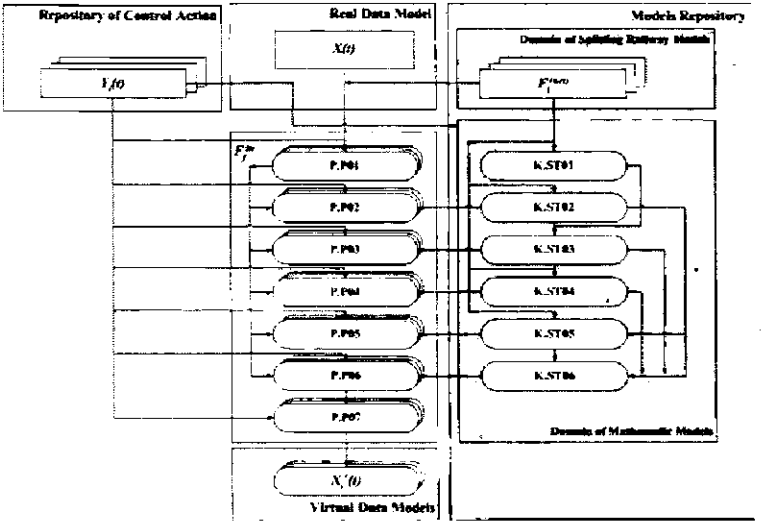


Figure 7. Conceptual model passenger streams analysis system

The suggested conceptual scheme of building virtual models using sole and independent from the railway network mathematical model gives opportunity to have several analyzed data models concurrently and to create quickly new ones. This approach widens the borders of the system application on other railways as well. And there is no need to rebuild the whole system for this, it will be only necessary to describe the network scheme of the corresponding railway and the managing impacts and to load new transactional data into the real data model in DW.

Application of virtual models together with strict system formalization allows to reduce the time of developing new reports and to guarantee their reliability.

3. Conclusion

The methods suggested by the author provide resolution of the following practical tasks:

- effective processing of huge volumes of data;
- providing data reliability in the process of data transformation in DSS;

- building and parallel application of a set of possible models on which basis different scenarios of the system behaviour are made without any perceptible increase of the stored data volumes;
- quick analysis of a big number of possible variants of the situation development with different combination of quantity and degree of outside impacts, system reaction to different managing affects and working out ideas about optimal system management in different situations;
- realization of the temporal database principles on the basis of the relational database in IS of railways.

The described methods have been tested on the Latvian railway and can be also applied in information systems of other business spheres.

In our further research we assume to carry out a more detailed formalization of the system and of some processes for practical realization of the system behaviour forecasts in the sphere of passenger transportations, as well as to evaluate the reliability of the yielded results.

References

[1] J. Kopitovs, V. Demidovs, N. Petoukhova. Use of Temporal Databases Principles in Information Systems of Latvian Railway. In: Proceedings of VII International Conference "TransBaltica 2002". June 12-14, 2002, Riga, Latvia. - p. 183-190.

[2] V. Demidovs. Use of Methods for the Increase of Data Reliability in Decision Support Systems of Latvian Railway. In: Transport and Telecommunication. Volume 4, Nr.1 – 2003, Riga, Latvia. - p. 16-22.

[3] J. Kopitovs, V. Demidovs, N. Petoukhova. Method of Temporal Databases design Using Relational Environment. In: Scientific proceedings of Riga Technical University – Computer Science: Applied Computer Systems. Series # 5, Issue # 13. Riga: RTU, 2002. - p. 236-246.

[4] J. Kopitovs, V. Demidovs, N. Petoukhova. Application of Virtual Data Models in Decision Support Systems. In: Abstract Book of the International Scientific and Technical Conference. April 17-18, 2003, Moscow: MSTUCA. - p. 181-182.

[5] J. Kopitovs, V. Demidovs, N. Petoukhova. Virtual Models in Forecasting Systems of Railway Transportation. In: Proceedings of the International Conference "Modelling and Simulation of Business Systems (MOSIBUS 2003)". May 13-14, 2003, Vilnius, Lithuania. - p. 265-268.

[6] J. Kopitovs, V. Demidovs, N. Petoukhova. Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport. In: Abstract Book of the Sixth International Conference on Computing Anticipatory Systems (CASYS'03), D.M. Dubois (Ed.). Liege, Belgium, August 11-16, 2003, Symposium 8. CHAOS, Institute of Mathematics, University of Liege, p. 8.

[7] Eugene A. Kopytov, Vasilijs Demidovs. Effective Access to Historical data in Temporal Databases. In: Conference Proceedings of the International Workshops on Harbour, Maritime and Multimodal Logistics (HMS & MAS 2003), Edited by Y. Merkuryev A. Bruzzone, G. Merkurjeva, L. Novicky, E. Williams. September 18-20, 2003, Riga, Latvia: RTU, 2003, p. 235-241.

[8] V. Demidovs. Virtual Models in Decision Support Systems of Latvian Railway. In: Proceedings of the International Conference "Reliability and Statistics in Transportation and Communication (RealStat'03)" – Transport and Telecommunication. Volume 5, Nr.2 – 2004, Riga, Latvia. - p. 25-35.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
139-150 P.

Development of a Complex Security System in Relational Databases for Railway Transport

Natalia Petoukhova

Doctoral (PhD) student, Transport and Telecommunication Institute
Lomonosova 21, Riga, LV 1019, Latvia
Ph.: (+371) 5834067, e-mail: natalia@ldz.lv

Supervisor of doctoral studies – professor Dr. habil. sc. ing. Jevgenijs Kopitovs

Abstract. The present paper gives the results of research in resolving problems of relational databases security for information systems in railway transport in the period from 2001 to 2004. The doctoral work is devoted to developing the methodology of building a complex system controlling access to relational databases in railway transport information systems. The advantage of the suggested system is possibility of realization of any security policy rules and account of semantics of the protected data. The system is based on the method of access control to the data at a level of tuples of the relation. The author suggests formal description and practical application of the complex access control task for different types of information systems – on-line, analytical and temporal ones.

Keywords: security, relational database, access control, security politics, restriction of access to a tuple; analytical, on-line and temporal systems

1. General description of the work

1.1. Actuality

The question of data security in information systems (IS) has been always actual for transport enterprises. Through the years of their activity the enterprise IS have generated a great number of valuable information resources. The majority of them have been committed to manage corporative database systems. IS data security depends above all on the effective organization of system security in a corporative database. And the more developed the information infrastructure of the enterprise is the more effective and secure the IS system security should be.

Latvian railway (LDz) today has a pretty well developed information infrastructure including more than one thousand registered users of the of the computing network of which only a few hundreds are making active use of the corporative database managing

common resources of the enterprise [1]. The number of users and their role in the system are constantly changing. In this connection the problem of providing information security of the multi-user database (DB) is becoming more and more actual.

The existing information systems security system is limited by the available data security standards, which do not allow complete realization of the enterprise security policy based on a great number of laws, norms and local orders. It is possible to illustrate this by the following tasks of LDz: system classifiers and codifiers, system of immovables stock-taking, working place of a stock cashier, registering computer periphery. Only for the above tasks security policy assumes different aspects of access differentiation, such as territorial and administrative division, time frames, functional roles and information confidentiality. And realization of these complex security policy conditions in real information systems seems to be a serious problem in spite of the fact that the question of relational data security has been largely considered by now.

1.2. Aim and tasks of research

The aim of the doctoral thesis is increasing the efficiency of security systems application for IS in railway transport by means of using new methods of controlling access to these systems' objects.

The following tasks have been set in the work:

- investigate demands for data security and the specificity of the existing LDz IS;
- evaluate standards existing in the sphere data security;
- work out methods restricting access to IS objects;
- work out a conception of security system integration with IS application task;
- work out methodology of building complex relational data security systems, which could be used in IS of enterprises with a developed information infrastructure;
- to apply the received methodology to different types of IS in transport – on-line, historical and analytical.

1.3. Scientific and practical value of the work

The scientific value of the work is development of the principles of building of flexible complex security systems, which are open to modification and which can realize all requirements of security policy. The advantage of the suggested system is its capability to be integrated with the protected data and the application sphere of the task as compared with the existing solutions, which present security system as a separate stage of an information system.

As an instrument of the security system realization the author suggests a method of giving access to data at the tuple relation level.

There have been suggested approaches to protecting data of three main types of IS – on-line, historical and analytical ones. The following tasks have been carried out to achieve the aim:

– There has been carried the analysis of state and factors influencing the LDz IS security and of demands for data security as well as the IS specificity.

– Temporal model of IS data in transport has been developed and methods of historical data security suggested.

– Method of virtual data models has been suggested for analytical systems.

Practical value of the work is application of the complex access control (CAC) methodology, which helps to increase security of relational systems' data and to make the security system a flexible instrument for restricting data access according to any conditions and demands of the enterprise security policy.

Approbation of the work. The results of the author's research have been published in 13 scientific papers (articles and report theses) [1-13] and reported at ten scientific and scientific-practical conferences.

2. Description of main research tasks

2.1. Role of security system in transport

At present, huge organizations like LDz possess a well-developed information infrastructure that contain various purpose IS as an integral part. These IS could be divided into 2 classes: on-line and analytic. On-line systems support an everyday life of an enterprise. As regards analytic IS, they are necessary for enterprise activity planning and evaluation.

Providing the access control to the on-line systems data is a matter of a primary importance, because of the fact that guarantees the effectiveness of the enterprise activities, depending on the data correctness and security. As far as on-line systems life cycle stored data are concerned, it is necessary to notice its further usage for prospective enterprise activities planning [1,2].

Analytic systems used as an information basis take the benefit of interacting with corporate data warehouse (DW) that keeps enterprise strategically significant resources produced from the on-line system data that have been stored for years, as well as gathered from the other sources. The value of the DW information resources includes the huge latent danger that is even greater comparing to the on-line IS databases. It is explained by the reason that allows to find out existing trends, dependencies and strategically valuable knowledge from the stored data, which may be used against the enterprise. At the same time this information potential has to be available for the enterprise analysts at different levels, as well as for external clients, business partners and commercial organizations.

In the on-line and in analytical type of IS tasks connected to the calculation of time factor and approach to "historical" data have obtained the largest topicality. Temporal data security requires the extraordinary approach due to the particular data organization

and due to the chronological orientation of security policy rules.

An IS interaction, tasks presence having mostly the temporal character and connected to this aspect security questions are presented at the Fig.1.

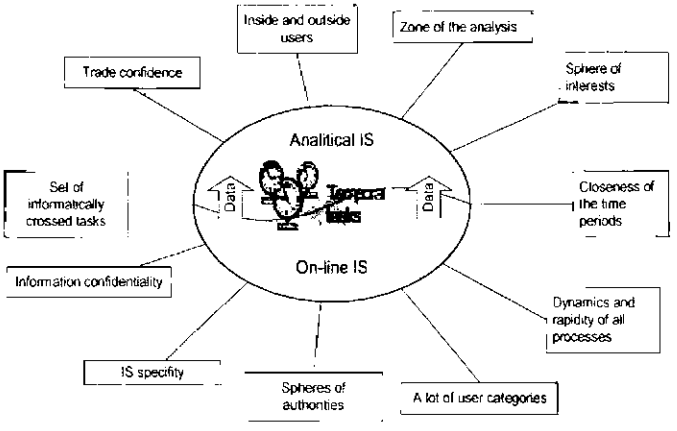


Figure 1. Security issues and IS interaction

The difficulty of the illustrated security conditions implementation for the transport IS is rooted in the fact that existing security issue standards are not designed for the type of this field-oriented approach.

2.2. Critical analysis of available approaches to data protection

The author has researched two of the presently available approaches to relational data access control – discretional and mandatory ones. The results of the research have been registered in the author's works [3,5,6].

Discretional access control (DAC) allows differentiating access between the named subjects and named objects [14]. It is included into the majority of commercial relational products and meets the majority of information systems, but is has some limitations in management flexibility.

SQL command managing GRANT privileges could serve an example of a discretional approach element GRANT command syntax looks as follows:

Operator of assigning object access authority | **GRANT**
Access privileges | **(INSERT, DELETE, SELECT, UPDATE)** (***column-name***) — *Access to selecting and modification can be restricted by some columns*
Object type | **ON TABLE**
Object name | ***table-name view-name***
TO (USER, GROUP)
User or user group name | ***authorization-name***
Allows to grant the privileges to other | **WITH GRANT OPTION**

From the command syntax we can see that access privileges for users (authorization

name) are associated with named information structures (table-, column-, view-name), which contain data but there is no connection with the data themselves. Security system does not know anything about their semantics. Privileges exist separately from the data, which from the point of view of security provision look somewhat depersonalised. Mandatory approach from this viewpoint is more advantageous than the discretional one.

Mandatory access control (MAC) defined access rights by comparing the security classification of the requested objects with the security clearance of the subject [14,15]. MAC presupposes obligatory and rather specific administrating of data security. Such approach is convenient for militarised and other tasks connected with super secret data. It has been developed particularly for these tasks. But for the majority of application tasks such need of clear administrating is too strict a condition.

Both in MAC and DAC security system is a detached element of an application task. MAC in this plane is more advantageous, as it has been mentioned above, since it allows restricting access to any object but not to the structure containing objects. But this approach also lacks connection with data semantics and, therefore, security system is not integrated with the application task. Besides it should be noted that some realizations of MAC are performed as closed systems. And it greatly limits transferability.

The above reasons explain limited application of MAC in real IS in railway transport, which security policy does not correspond to the mandatory approach model.

Application of the above classic approaches does not solve the task set and new methods of solving the problem are needed, which could successfully meet the requirements of IS security in transport.

2.3. Working out principles of building a complex security system

Referred author's paper [13] is devoted to complex security system requirements development. Further basic of these requirements are formulated.

– The level of access to the data is defined by the data and depends on their semantics, i.e. the data participate in the process of restriction of access to it.

– System of access control may function as an integrated part of IS and follow to the actual IS access subject and object activity rules. It means that there exists the possibility for the security system to function without a direct administrative interference.

– The system should allow realization of any security policy and its any rules.

– There should be a possibility of accountability processes arrangement.

– User may have the different access levels to the various operations for the same data sets. Read-, write- and modification accessible data arrays of the same information object may vary, e.g.

Access to the data should depend on their semantics, instead of structural elements in which they are located. Hence, we are facing the problem of differentiation of access inside the named information objects, such as the relational table. The method of access control to the data at a level of tuples of the relation in relational systems has been considered in work of the author [6].

Table data are considered to be the security object of the proposed method. By means

of applying this method one can restrict the table data access in the way, which allows user to get the access to one of table records, remaining another one untouchable. Access restriction or allowance takes place on the basis of the special row security labels, user identifier and defined security rules. As a data *security label marker* will be regarded the some marker that is present in data and is pointed to as a data access criteria.

The method is illustrated at Fig.1. *User* requesting *data* from the table is going through the security system layer that is defined by *access rules*. According to these rules, user gets the access only to the selected records that are marked with a dark colour. User-accessible records are determined by values of *Label 1* and *Label 2*, corresponding with data usage *user properties*, marked with the appropriate labels.

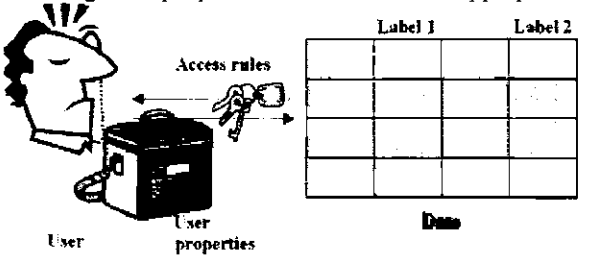


Figure 1. User access to the relational table

Technically, data relational access differentiation is encouraged by creation of a separate view (virtual table) that is a tool of any user interaction with a source table and two triggers, aimed for limiting the modification and deleting access.

Formalization of a task of restriction of access to a tuple of the relation

In the formal task description author has exploited Boolean algebra and array theory methods. This aspect is described in the work [13].

For the task formalization we shall determine groups of sets that participate in the process of the CAC. Consequently, tuples with a restricted access, security system specialized objects, as well as user request elements will be described.

The selected tuple with a restricted access. The tuple structure t of relation T is presented as: $t = \{A, L\}$,

where $A = \{a\}$ – set of tuple attributes, not interfering with its access;

$L = \{l\}$ – set of tuple security labels (set of tuple attributes, defining its access), having power m.

Security system specialized objects. Let us introduce the following sets:

$U = \{u\}$ – set of users;

$O = \{o\}$ – set of possible access operations to a tuple;

$Q = \{Q_1, Q_2, \ldots, Q_m\}$ – domain sets of security label values defined for each label $l_i \in L$, $i = \overline{1,m}$, where $Q_i = \{q^{(i)}\}$ – domain of security labels values, which is

defined for label $l_i \in L$; $q^{(i)}$ – possible value for i-th label;

$S = \{S_1, S_2, \ldots, S_m\}$ – domain sets of user properties, interfering with access to label $l_i \in L$, where $i = \overline{1,m}$;

$S_i = \{s^{(i)}\}$ – domain of user properties values, interfering with access to label $l_i \in L$; $s^{(i)}$ – possible value of user property.

$R = \{R_1, R_2, \ldots, R_m\}$ – set of collection of access rules to labels from L, where $R_i = \{r_1^{(i)}, r_2^{(i)}, \ldots, r_{k_i}^{(i)}\}$ – collection of access rules to label $l_i \in L$;

k_i – number of access rules to label $l_i \in L$, $i = \overline{1,m}$.

The rule $r_j^{(i)} \in R_i$ regulates access of users with property $p \in S_i$ to label $l_i \in L$ that has value $q \in Q_i$, where $i = \overline{1,m}$, j - number of rule.

The structure of rule $r_j^{(i)} \in R_i$ is resulted below:

$r_j^{(i)} = < p, q, o >$, where $p \in S_i$, $q \in Q_i$, $o \in O$, j – number of rule.

Elements of query of the user to the data of a tuple:

$u \in U$ – user that has sent the query;

$o^{(u)} \in O$ – operation of query of user u;

$P^{(u)} = \{p^{(u)}\}$ – set of properties values of user $u \in U$ that has sent the query;

$P_i^{(u)} \subseteq P^{(u)}$ – set of user properties values, interfering with access to label $l_i \in L$, $i = \overline{1,m}$, i.e. we has $P_i^{(u)} \subseteq S_i$;

$L(u,o) \subseteq Q$ – set of labels values series that are regarded to be user-accessible for performing operation $o^{(u)} \in O$ for the tuple t, which is defined as $L(u,o) = \{l_1(u,o), l_2(u,o), \ldots, l_m(u,o)\}$,

$l_i(u,o) \subseteq Q_i$ – set of labels values that are accessible for user $u \in U$ for performing operation $o^{(u)} \in O$ for the label $l_i \in L$, $i = \overline{1,m}$, which is calculated as follows: $l_i(u,o) = \underset{F_i, o^{(u)}}{P_i^{(u)} \times R_i}$, where $l_i(u,o) \in L(u,o)$, $i = \overline{1,m}$, $\times$ - the operator of the Cartesian product, F_i – collection of security politics functions, that is defined for label $l_i \in L$.

The calculation of tuple accessibility. A sets of tuple accessibility facts is Boolean vector X, which is defined by check of occurrence of tuple labels set L in corresponding sets of labels accessible to the user:

$$X = L \subseteq L(u,o), \text{ i.e. } x_i = \begin{cases} true, & \text{if} \quad l_i \in l_i(u,o), \\ false, & \text{if} \quad l_i \notin l_i(u,o) \end{cases}, \; i = \overline{1,m}$$

Then the y - fact of accessibility of a tuple is defined as follows:

$y = B(X)$, where B is Boolean function. In other words, a set of values y is $\{true, false\}$. In case of validity of y, access to the tuple $t = \{A, L\}$ is permitted otherwise the access is restricted.

2.4. The solution for the logging task

The trusted system should fix all events concerning security. Conducting reports should be supplemented with audit [15]. In case of total continuous events logging, the registered information volume obviously will grow too fast, and its effective analysis will become impossible. Security system expects the existence of sample logging means aimed for users (paying attention only to suspicious ones), its properties and attributes, as well as for the operations and reference time.

For the logging task formalization let's enter the following designations: $O_Z \subseteq O$ – set of logging operations, $A_Z \subseteq A$ – set of logging attributes, $U_Z \subseteq U$ – set of logging users, $P_Z \subseteq S$ – set of logging user properties values, $T_Z = \{\tau^{(Z)}\}$ – set of logging time periods.

Logging process is started up in case of observance of the following condition:

$$\left(o^{(u)} \in O_Z\right) \vee \left(A^{(u)} \subseteq A_Z\right) \vee \left(u \in U_Z\right) \vee \left(P^{(u)} \subseteq P_Z\right) \vee \left(\tau^{(u)} \in T_Z\right),$$

where $o^{(u)}, A^{(u)}, u, P^{(u)}, \tau^{(u)}$ – corresponding sets of an environment of user query.

3. Practical application of the developed system

CAC can find practical application in any multi-user IS assuming differentiated access to objects of the same type. Let's consider application of access control system for the main types of IS in transport – on-line, historical and analytical ones.

3.1. The complex access control for on-line system

The account of hardware maintenance LDz is an indicative *example of a task of the CAC for the on-line system* operating general corporate resources.

Security policy of account task is formulated as follows:

– access to the data of hardware devices is defined by territorial and administrative spheres of users authorities and it is possible, if corresponding attributes of the device are laying in this spheres;

– the hierarchy of departments is taken into account for the definition of authorities;

This is an indicative example, because the access control is not the clearly abstract security policy module. It is strongly correlated with actual state of enterprise actions, namely depending on the structural departments changing hierarchy, as well as on user belonging to one or another department. These data are changing permanently during the life cycle of the enterprise. Security system takes into account the abovementioned changes immediately by means of correcting processes of data access.

In the author works, general tools of security system realization are SQL standard means. The system is fulfilled at DBMS IBM DB2 UDB v.7.2 platform.

Part of DB schema of the example is shown in Fig.2. DB contain the following tables: HDevice - the table of hardware devices to which it is necessary to supervise access: Users - the list of all users of system with the instruction of their administrative subordination (attribute Str); Str_Hrhy – the hierarchy of departments. The listed tables contain only the data from a subject of task of the account of hardware maintenance, and at them there are no specialized elements of security system. However, the security system uses some of these data.

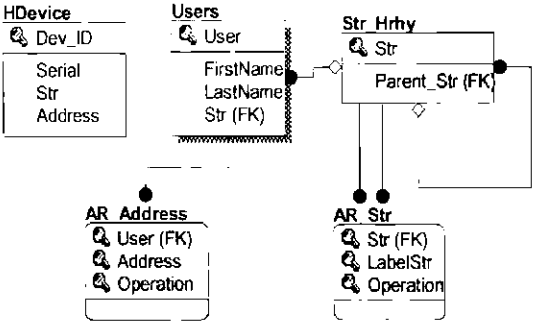


Figure 2. ER model of database

The specialized tables of security system are: AR_Address - territorial privileges of users access; AR_Str- administrative differentiation access rules.

The formal description of the given task is described in the work [13] of the author. The formula $y = B(X)$, determining access of the user to a tuple of Hdevice relation is deduced. Each tuple of this relation corresponds to the separate hardware device. *Function of accessibility of a record for the device* in terms of relational algebra looks as follows:

$$y = \sigma_{x_{Str} \vee x_{Address}}(T),$$

where $x_{Str} = Str \subseteq \underset{Str}{\pi_{Str}} \sigma_f (AR\_Str \underset{Parent\_Str=Users.Str}{\times} Str\_Hrhy \times Users)$;

$$x_{Address} = Address \subseteq \pi_{Address} \sigma_f (AR\_Address);$$

$$f = (Operation = OPERATION) \vee (User = USER);$$

OPERATION – access operation; *USER* – system variable, specifying the identifier of the user initiated access to the data.

The received expression $y = B(X)$ is simple enough for presenting *in query language SQL*:

```
y = Hdevice.Str IN ( SELECT a.Str from AR_Str AS a INNER JOIN Str_Hrhy AS s
                        ON a.Str=s.Str INNER JOIN Users AS u ON s. Parent_Str=u.Str
                     WHERE u.User=USER AND a.Operation=OPERATION )
   AND
   Hdevice.Address IN ( SELECT a.Address from AR_ Address AS a
                        WHERE a.User=USER AND a.Operation=OPERATION )
```

The deduced formula of calculation of the device accessibility record $y = B(X)$ in an obvious way of presentation is set for developed views and triggers in the security system. At change of a security policy and, hence, completions of function of calculation $y = B(X)$, it is easily replaced with the new formula without change of the general architecture of security system.

3.2. The complex access control in historical IS

The application of the proposed methodology for creation of security system of a *temporal IS* appeared to be rather valuable. Some of the author's works [4,7,8] are devoted to the temporal systems investigation.

For the organization of the temporal data in a relational DB it has been offered *temporal data model with abstract object identifier*. Designing of the given model and its use is considered in works of the author [4,7,8].

Different tasks of user actions restriction in time while working in the temporal DB were preceded by the author. Some of tasks are dealing with the information processing prohibition of user actions for the previous time and future time periods. It is important to note, that as a rule, it is required to differentiate chronological access of access operations. User may be given the possibility to analyze information for the period of previous two years, permitting data update only for the previous month, e.g.

3.3. The complex access control in analytical IS

The entire system analysis at the railway has to include the current and previous evaluation of activity, as well as its forecasting. It is proposed to take the benefit of the modern data warehouses usage, keeping large amounts of information. DW consolidate data, which are generated by on-line IS and external sources. Multi-user appropriation of DW includes a number of problems, closely connected to confidentiality of consolidated data. The general reason for this fact is that users are the analysts of both internal and external commercial structures. As a rule, in systems of analysis it is necessary to select *limited analysis zones* for each analyst. Analytical systems, DW and relevant questions research results are stated in the author works [1, 2, 9,10,11].

Traditional approaches of various analytical task solution, like a tasks of forecasting in the DW, are rooted in exchange of a real data with predicted or in cloning of a part of a source DW for every analyst aiming its further modification. Such approaches have

shortcomings, connected with DW blocking for the forecast time period, the complexity of further DW recovery or the surplus disk space.

In the several author works [9,10,11] it is proposed the new approach for DW creation in the decision-making and forecasting systems. The approach sets its objective to create virtual system models that do not require disk space, because of additional location only of metadata and Models Repository that describe the set of virtual models, constructed upon the real data model basis on the one hand and influence control vectors, from the other hand. Alongside with influence control vectors the particular meaning in the model have *data security access control elements.*

4. Conclusion

The suggested access control system gives users their own view of a corporative database without exceeding the frame of a specially allocated database subschema. The system is realized by standard SQL means: triggers and views. The system mathematical apparatus is relational algebra. CAC task is rather easily formalized.

CAC has found its application in LDz information systems and allows resolving a lot of problems connected with complex data security. The present research has shown its reliability and flexibility of application. The suggested approach can be applied to different types of tasks, both on-line and analytical ones. It has also proved its efficiency in working with temporal databases.

This method of access control can be also used in other IS since it is based on the condition of adjusting to any corporative rules. And using only standard means in complex data control realization makes possible building a CAC on the basis of actually any modern DBMS, i.e. the CAC system is transportable.

The author's further research is aimed at improvement of IS security system in transport and modeling access control elements. Special attention will be given to increasing the efficiency of security system in analytical tasks, in tasks of forecasting railway enterprise activity in particular.

REFERENCES

The author's publications

[1] E. Kopytov, N. Petoukhova, V. Demidov. Methodology of Huge Data Volume Processing System Development for Analysis of Latvian Railway Passengers Transportation. *In: Proceedings of VI International Conference "TransBaltica 2001"*, June 7-8, 2001, Riga, Latvia. p. 201-208.

[2] Петухова Н. Методология построения системы обработки больших объемов данных. *Konferences materiāli/ Informācijas tehnoloģija: zinības un prakse. VI konference, 28.nov.2001.g., Rīga, Latvija,.- lpp.65-70.*

[3] N.Petoukhova, Increase of safety of data in relational database for records level, *Программа и тезисы/ Научно-практическая и учебно-методическая конференция "Наука и технология - шаг в будущее", Институт транспорта и связи*, 2-3 мая 2002, Рига, Латвия. - с. 26.

[4] Е.А. Копытов, В.В. Демидов, Н.Ю. Петухова. Использование принципов временных баз данных в информационных системах на Латвийской железной дороге. *In: Proceedings of VII International Conference "TransBaltica 2002"*, June 12-14, 2002,Riga,Latvia.-p.183-190.

[5] Н.Петухова. Детализированный доступ к данным, как средство повышения безопасности информационных систем. *In: Proceedings of the International Conference "Reliability and Statistics in Transportation and Communication (RelStat'02)"*, October 17-18, 2002, Riga, Latvia. - p. 73-74.

[6] Н.Петухова. Метод обеспечения доступа к данным реляционных систем на уровне строк отношения. Transport and Telecommunication. Volume 4. Nr.1-2003, Riga, Latvia.-p.36-42.

[7] J. Kopitovs, V. Demidovs, N. Petoukhova. Method of Temporal Databases design Using Relational Environment. *In: Proceedings of the 43rd International Scientific Conference of Riga Technical University*, Riga, Latvia. - p. 19.

[8] J. Kopitovs, V. Demidovs, N. Petoukhova. Method of Temporal Databases design Using Relational Environment. *In: Scientific proceedings of Riga Technical University - Computer Science: Applied Computer Systems*, Issue # 13. - Riga: RTU, 2002. - p. 236-246.

[9] Е.А. Копытов, В.В. Демидов, Н.Ю. Петухова. Применение виртуальных моделей данных в системах принятия решений. В кн.: Гражданская авиация на современном этапе развития науки, техники и общества. Тезисы докладов Международной научно-технической конференции, посвященной 80-летию гражданской авиации России. 17-18 апреля, 2003, Москва, МГТУГА, с. 181-182.

[10] J. Kopitovs, V. Demidovs, N. Petoukhova. Virtual Models in Forecasting Systems of Railway Transportation. *The international conference: Modelling and Simulation of Business Systems (MOSIBUS 2003).* Vilnius, Lithuania, May 13-14, 2003. - p. 265-268.

[11] J. Kopitovs, V. Demidovs, N. Petoukhova. Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport. *Sixth International Conference on Computing Anticipatory Systems: CASYS '03.* Liège, Belgium, August 11-16, 2003. http://www.ulg.ac.be/mathgen/CHAOS/CASYS2003.htm

[12] N.Petoukhova. Principles of Development of Complex Security Systems in Relational Databases on the Example of Latvian Railway. In: Proceedings of the International Conference "Reliability and Statistics in Transportation and Communication (RealStat'03)", October 16-17, 2003, Riga. Latvia. - p.28-29.

[13] Н.Петухова. Принципы разработки комплексных систем безопасности в реляционных базах данных на примере Латвийской железной дороги.. In: Proceedings of the International Conference "Reliability and Statistics in Transportation and Communication (RealStat'03)". October 16-17, 2003, Riga, Latvia. – (in printing).

Other publications

[14] Козленко Л., «Информационная безопасность в современных системах управления базами данных», КомпьютерПресс 3, 2002

[15] Галатенко В.А., курс лекций «Основы информационной безопасности», http://publish.abitu.ru/courseslibrary/ims/c_yfksu/secbasics/lectures/lecture+5.esp

[16] Алан Р.Саймон. Стратегические технологии баз данных:Менеджмент на 2000 год. Пер. с англ./Под ред. М.Р.Когаловского.-М.:Финансы и статистика,1999. 402-405с.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
151 -154 P.

A Similarity Retrieval Algorithm for Natural Images

Natalia Vassilieva, Boris Novikov

St.Petersburg State University, Russia
vnat@nv10381.spb.edu, borisnov@acm.org

Abstract. Content-Based Image Retrieval (CBIR) has become one of the most active research areas in the past years and still is known as a difficult task. Using a retrieval system is generally frustrating for users, due to a gap between low-level features managed by system and image semantics. We propose in this paper a general point of view for introducing a bridge between the user and the system. This includes a textual query, image features, based on visual perception models, a relevence feedback.

Keywords. CBIR, Image retrieval, Content analysis, Visual perception

1. Introduction

The use of images for illustration has always had a wide distribution in human activities. Thanks to recent technological advances the existence of large digital image databases became possible. The growing size of the contemporary image collections has created the necessity of image retrieval systems. This reinforced the efforts of researches and leaded to appearance of various approaches. The earlier retrieval methods are based largely on the semantic keywords attached to the images. This can be done either by manual annotation or by automatically extracting the keyword from the context, when it exists. The first one is subjective and very time consuming, the second one is not always possible. More recent works propose automated Content-Based Image Retrieval (CBIR), based on the "low level features" (color, texture, etc.) extracted from pixel values.

The preferred mode of querying in image database is semantic. For example, we might search for images of a road in a forest. To satisfy such a query, the system must be able to recognize roads and forests in the images. But this level of interpretation in CBIR systems is still out of the question. As the system can estimate the similarity of images based on their low level features, there is an important "semantic gap" between the image features, which have been extracted, and the semantics of the image.

Most widely used kind of query in CBIR systems is a query by example (QBE), formulated by providing an example of a similar image. The system returns a set of images estimated as similar by image features.

2. Related work and motivation

Literature shows a huge amount of various techniques that have been applied to CBIR in the recent years. (In [4] you can find a good review of them.) The great number of different approaches can be explained by a wide variety of application domains and by the fact that the existing CBIR systems are still not as efficient as the user want them to be.

Typically, a CBIR system finds images from a large data collection that visually match to a given query. The most systems propose to define the query by providing one or more example images. Another possibility is to provide a rough sketch of the desired image [2]. But both of the ways leads to the inconsistency between the semantic query that the user has in mind and its description which makes it hard for the user to specify the query and for the system to return the correct images.

Existing approaches based on treatment of low level features can be classified by:

- considered features,
- used data models,
- applied similarity metrics.

Among the features the color is the most extensively used. It's meaningful in human perception, easy to extract and robust to noise, scaling and rotation. Majority of data models for representation of color features vary between different kinds of histograms [8, 9, 5] and statistic models of color distribution. Texture features are also widely used in image retrieval. For texture analysis [2, 3, 6] use wavelets, other approaches use synthesized banks of filters for texture extraction (see [10] for a full review). Shape features attract less attention of scientific community. Retrieval by shape is useful only in specific collections (e.g. items on a homogeneous background or geometrical images). See [1] for review of shape representation and retrieval approaches.

Many systems provide the possibility to combine or select between one or more models, based on different features. But they typically process the features independently and use several indices, which can increase both space and time requirements.

Among more recent works there are those that propose a kind of "semantic bridge" between the user and the system. The widely used mechanism of "relevance feedback" ([5], for example) takes into account user's satisfaction by iterative process of user-system interaction. During the retrieval process, the user high-level query and perceptual subjectivity are captured by dynamically refined queries based on the user's feedback.

3. Problem definition

We consider a database of natural images, where no additional semantic information about images is available. Research is performed in the following directions:

- provide a textual way of query definition, which is more clear and intuitive to the user than the query by example;

- select the most meaningful features for natural images and propose the unique model for its representation;
- design of indexing algorithms, that agree with human visual perception;

use clustering to reduce the number of images to process in the image retrieval phase and accelerate the retrieval.

4. Proposed solution

The complete image storage and retrieval process is divided into three phases. During the first one we define the training image set, which is expected to be much smaller than the image database itself. This set is divided into the groups of similar images by performing the classification task. For each of the groups we compute an average feature-vector from feature-vectors of group images. Feature-vector for an image is computed from its low-level features. We describe each group also by textual keywords.

The second phase serves to compute feature-vectors for the rest set of images and perform data clustering in order to the distance between an image feature-vector and average feature-vectors of the groups.

In the third phase of image retrieval we compare textual query of the user with textual descriptions of the clusters and define the most suitable cluster of images. Predefined number of images randomly selected from this cluster and returned to the user. After that the iterative process of "relevance feedback" is used: in each iteration user marks right and wrong images among the returned set. The system uses this information for query refinement.

5. Conclusions

The paper describes general ideas for centered user approach for image retrieving. These are the first steps, which are very promising and must be developed to obtain further results.

References

[1] Fan Shuang. Shape representation and Retrieval Using Distance Histograms. *PhD thesis, University of Alberta.* October 2001.

[2] Jacobs C. E., Finkelstein A., and Salesin D. H. Fast multiresolution image querying. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp 277–286,* Los Angeles, CA, August 1995.

[3] Kingsbury Nick. Image processing with complex wavelets. *Phil. Trans. Royal Society London A, vol. 357, pp. 2543-2560.* 1999.

[4] Rubner Y. Perceptual Metrics for Image Database Navigation. *PhD thesis, Stanford University.* May 1999

[5] Rui Yong, Huang Tomas, Ortega Michael, Mehrotra Sharad. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Video Technology.* 1998.

[6] Siggelkow Sven. Feature histograms for content based image retrieval. *PhD thesis, Albert-Ludwigs-Universitat Freiburg im Breisgau*. 2002.

[7] Simoncelli Eero, Portilla Javier. Texture characterization via joint statistics of wavelet coefficient magnitudes. *Proceedings of Fifth International Conference on Image Processing, vol I*. Chicago, IL, 4-7 October 1998.

[8] Stricker Markus, Orengo Markus. Similarity of color images. *In Storage and Retrieval for Image and Video Databases III, SPIE 2420, pages 381--392, San Jose, CA*. February 1995.

[9] Swain M. J. and Ballard D. H. Color indexing. *Intern. Journal of Computer Vision 7(1), pp. 11-32*, 1991

[10] Tuceryan Mihran, Jain Anil. *Texture analysis. The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248*, World Scientific Publishing Co., 1998.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
155–163 P.

Temporal Storage and Representation of Integrated Health Data

Tore Mallaug[1, 2], Kjell Bratbergsengen[1, 3]

[1]The Department of Computer and Information Science,
Faculty of Information Technology, Mathematics and Electrical Engineering,
Norwegian University of Science and Technology (NTNU)
[2]PhD student and contact author
[3]Advisor
{torem, kjellb} @idi.ntnu.no

Abstract. The demand for timely, accurate health data is continuously growing (e.g. [4]). The increasing volume of data collected from different sources creates new needs for a future health information system representing a totally intergraded electronic health record (EHR). The integration of data makes it easier for citizens to collect and control their own personal health data. The database plays an important role in such a future system. All kinds of personal health data must be stored and represented for a long-term access. For this purpose a simple temporal object model is introduced. Any stored objects are given time stamps on a linear time axis. Temporal / historical perspectives on the long-term stored data can be generated by searching these time stamps. XML technology is used as an example of the content of the object data and their corresponding schemas. Electronic referral can be used as a case for the health care services.

Keywords. Long-term storage, temporal data model, health informatics, XML

1. Introduction and research objectives

In a modern health care system the need for data exchange of personal health data is increasing. There are numerous reasons for this new demand. In a modern society citizens move and travel more frequently, and then they need to have their personal health data available where they stay or live at the moment. In Norway, citizens have a legal right for a free choice of a hospital, and a right for inspection of all their personal health data stored by different health care providers (today this is difficult to achieve since the data is stored in different local databases all over the country). These trends ask for national database solutions, and in a longer perspective, international solutions. From a database point of view, a future solution of a common integrated database is an alternative to a direct message passing between heterogeneous information systems. This scenario can include database benefits, such as better data quality, better data availability, storage optimization, back up and logging administration, and common access control. We are going from a message passing system to a data sharing system.

The research objectives of this PhD work can be separated into two sub parts:

1. To evaluate the future database solutions for a common integrated EHR – Electronic Health Record [15].
2. To explore the long-term temporal data representation of personal health data and mappings of different versions of these data in a time space.

The second part can be used in an implementation of the first part. The EHR is a wider, extended definition of the CPR (Computer-based Patient Record, e.g. [1] [4]) that includes all kinds of personal health data and is not only limited to, for example, treatment at a hospital.

Our contribution is to look how future EHR can be stored and how the data can be represented from a database point of view. We also focus on a long-term access of historical health data, since personal EHR must be accessible at least for the whole person's life (100 years). Both the storage technology and the data representation will continuously change during such a long time period. This approach is different from traditional work on CPR since we do not constrain ourselves to commercial systems of today, by the situation of organizational problems in the health care sector or by legal limitations. Thus, we hope to offer applicable future solutions.

As a demonstrator we use electronic referrals between service providers, for example the process of local doctors referrals to a regional hospital. In Norway a national standard for electronic referrals defined in XML Schema is being developed [8].

2. Related projects and present realities

2.1. Database solutions in the health care sector

The fully integrated database solution for personal health data has not been yet developed. So far, the operating CPR's are not at the EHR level (e.g. [9]). A future personal "virtual health record" activated by linking records on the Internet is mentioned by [13], but no database solution is related to it. Some national databases exist, like in Iceland [6], but these databases store only "anonymized" health data. In Norway, many local databases exist, some include sensitive personal data, and some health registers with "anonymized" data meant for statistics and public information. The chance for redundancy between local databases is high - the patient has to repeat the same information for registration, if visiting different health service providers. It is also a high risk for bad data quality, for example lack of update and control and even worse; lack of a common vocabulary. In practice, a patient has no chance to collect and control the access and use of all his/hers registered personal health data.

Database solutions are typically linked to the commercial software for the health care. A common middle layer service does not exist; all communication between systems has to be done by direct message passing between local applications. for example EDI or e-mails. There are both national and international projects on XML standards for electronic message exchange (e.g. [2]) in health care.

Data security issues are essential in the health data context. These include the identification of the end users by using a smart card or by biometrical solutions. We have chosen not to focus on the security, since data security is being fastly developed and the situation will be different in, say 10 years from now, when the EHR can start to be a reality.

2.2. Schema evolution

For schema evolution we have seen approaches, like for DTD's (e.g. [3] [12]). Mappings are typically related to the creation of common (global) integrated schemas. Some of the mappings are not "total" in the sense that they only map subparts of the original data content in different local schemas. Traditional schema evolution also typically maps only in one direction. The concern is that the mapping process is a one-time operation, meaning that all client programs have to be re-implemented at the same time the new mapping is released.

3. Design

So far, we have worked on a long-term temporal data representation of the health data. Below we describe some of the design ideas for the implementation of a common middle layer including the temporal object model.

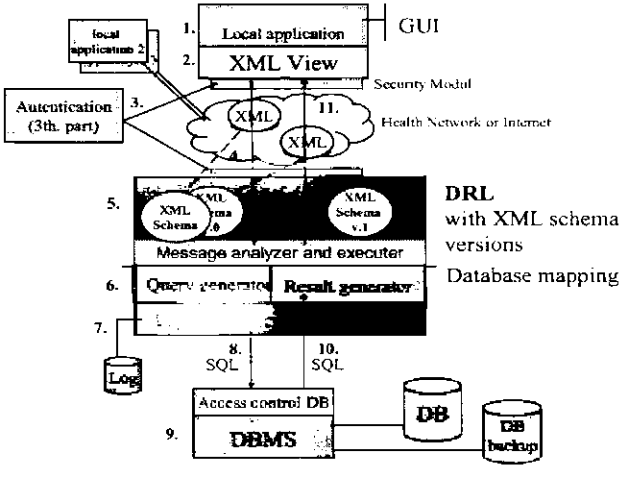


Figure 1. Three-level architecture

3.1. 3-tier client/server architecture

A client/server design for the overall system is presented in Figure 1. It consists of a middle layer which we called the Data Representation Layer - DRL, an underlying DBMS solution and a local client program. The DRL offers common services to heterogeneous client programs, as a set of tools, such as a mapping

generator, message validation and query execution, logging and access control. The architecture does not set any restrictions on the underlying database system if it is a mapping between the database and the DRL and if querying can be handled efficiently (for example when querying data in emergency situations).

3.2. Temporal object data model

The DRL does not include any global integrated total schema for the underlying stored data as mediator's layers ("canonical" schema) typically have (e.g. [7]). Clients can create such a view by introducing a schema standard for the purpose. The implementation of object-oriented methods (behavior) is also left to the clients. Our model is only for *data content* representation. The general goal is to make the model flexible for future changes, as is the framework for our temporal data representation. Later on we can examine if there is any existing data model that fit to our final requirements.

3.3. Temporal reading and writing of data

The goal of the data model is to represent data content in a temporal environment for a long-term use. Data content can be like elements in XML, and its metadata. An object includes a set of *elements* related to a well-known schema version. The schema itself is described by its DDL (Data Definition Language) and stored as a special object type called schema object in DRL. The model accepts versions of data and schema objects and relationships between them. All stored objects must be related to a set of time stamps (e.g. [14]). This set includes a fixed time stamp called *object date* which stores the transaction time when the new object instance was created / inserted into the database. A given object version instance is identified by its *object date* and a universal unique object identifier (OID).

For schema objects, each single version reflecting its time's (medical) development, practice, needs, and function. The result is a set of evolving schema versions represented in an ascending chronological order. The option not to delete objects gives a unique possibility to re-construct the situation of the time of selected periods / intervals. Reading / retrieving objects in a temporal perspective has meaning only when using a special time stamp we called the *read date*. If following a linear time axis, as illustrated in Figure 2, the *read date* can be seen as an accurate time line on this axis. The reader may then look at "the world" as it was expressed at that given historical date. The basic philosophy is that any object must be seen in relation to its *object date* plus the *read date*. This approach represents nearly unlimited ways for local client applications to query and view data versions both forward and backward in time. Some examples of reading versions are illustrated in Figure 3. A client application can read data content from one or many object versions according to one or two *read dates* (time_read in the Figure 3).

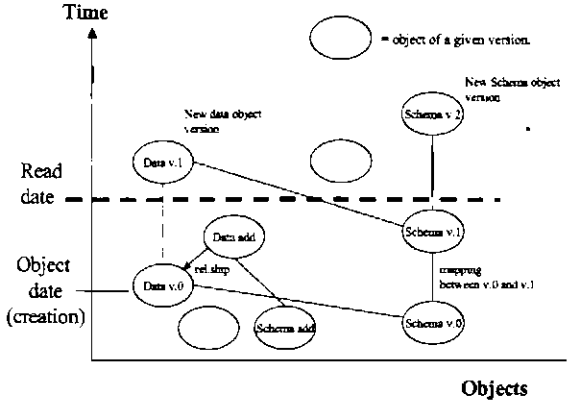


Figure 2. Versions of data objects and schema objects by time.

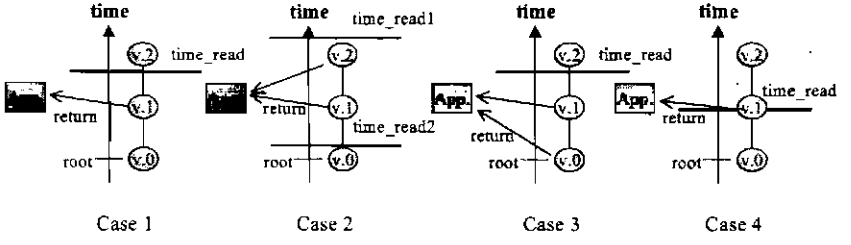


Figure 3. Cases of temporal reads of object versions related to given read dates.

3.4. Evolutions of schema versions and standards

There are 2 types of evolutions of schema standards:

1. Schema developing inside the same schema standard, like the evolution of DTD's from one DTD to another. In such cases, both schemas are assumed to be compatible, even if there can be several versions of the same schema standard.
2. Developing from one schema standard to another (newer) standard. Such cases can be split into:
   2.1. Evolution between two compatible standards, like from DTD to XML Schema.
   2.2. Evolution between none-compatible standards, like from the relational data model (for example represented in SQL) to XML.

We set a restriction saying that any element in a given schema must have one and only one mapping to a newer schema. A *total mapping* for a schema is defined as including all the schema's elements and to always providing the same single result that is total and absolute, not partial or accidental. In general, mappings from a given schema object version can be implemented as: 1) Mapping between two schema object versions - evolution between versions. 2) Mapping from one old schema object to one or many new schema objects - evolution between old and new

schemas. 3) The old schema object is replaced or expired without any mapping – that is, a paradigm shift occurs. We attempt to define the rules for a *total mapping* and set up a XML Schema for the representation of such mappings.

3.5. Mapping object

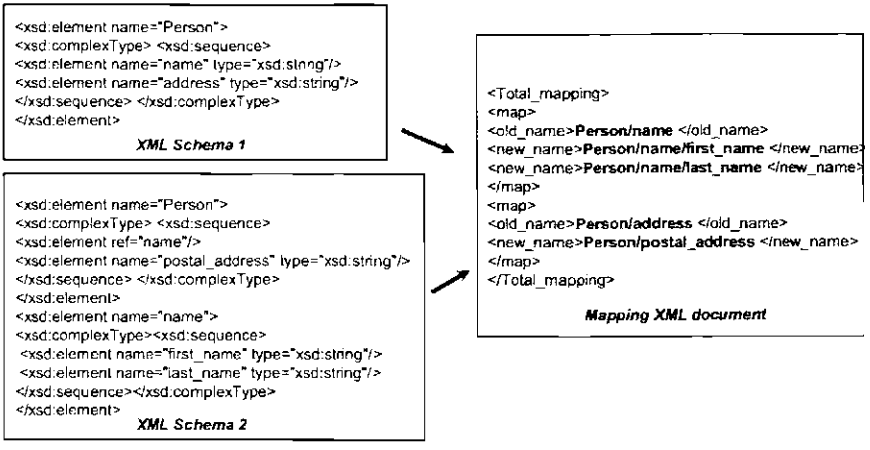


Figure 4. An example of mapping between two schema versions using the XML Schema standard.

Figure 4 shows an example of a mapping between two schema versions, representing a simple *total mapping* in XML. The mapping XML document is an example of a mapping object used to store data sets about the mapping. The content can be extended to include a type converting. Still the mapping deals with syntactic differences only. To implement temporal reads as the ones in Figure 3, we also need more sophisticated data about the heterogeneity of schemas. Schema differences can be divided into three levels:

- *Syntactic level* : A mapping as the one in Figure 4 is limited to only syntactic, or technical, differences between schemas. In some cases, such mapping can be a one-to-one relationship from one schema to the other, though this is not necessary true in both directions of a mapping. An example is the MedCom organization [10] in Denmark which offers a one-to-one mapping between EDIFACT and XML messages used by health care services.
- *Conceptual level* : A mapping which tries to represent semantic, or conceptual, differences as well. For example, a schema which is based on an object oriented data model is typical semantically "stronger" than a schema based on the relational data model. It is difficult to represent such cases only by syntactic differences, since then semantic knowledge can be lost during the mapping.
- *Semantic (ontological) level* : This level investigates semantic heterogeneity (e.g. [16]) linked to the view of the reality, that is, the state of the world that the schema end users have. Such an ontology is not based on

conceptual differences in the schemas but rather on people's perception of the data content [16]. If a new and an old schema are based on two different (sub-) ontologies, the mapping can be wrong even if it is conceptually seen as right. If the mapping object can store data about such differences, the data can be used to understand (interpret) ageing ontologies when reading old historically health data.

Mapping rules can be set up and stored in the mapping object for all the three levels. To represent differences in ontologies we probably need to link mapping rules to formal logic.

4. Implementation and results

A simple prototype of the 3-tier client/server architecture was implemented using Java and SOAP in an interactive waiting-list system[1]. However, waiting list data is not a good example of personal health data since the data is not considered to be a part of the CPR, rather a part of the PAS (Patient Administration System) at hospitals. We are now using electronic referral as a case since it is partly related to the waiting list systems. As example we can map between beta versions of an upcoming Norwegian XML standard [8]. Since we do not focus only on XML only, we can also look at the mapping of the Norwegian XML standard and a Danish EDI standard [10]. This can make it possible to send a Norwegian referral to a Danish hospital (in Norway some patients are sent abroad for their medical treatment).

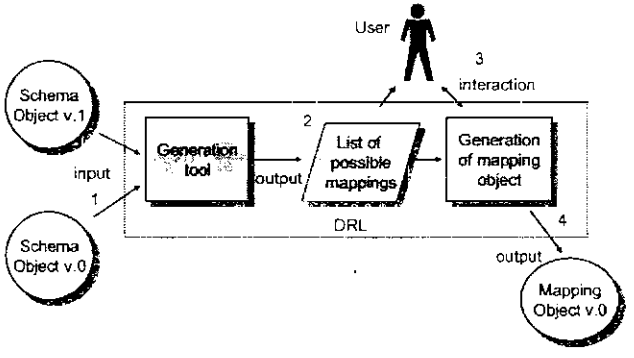


Figure 5. The Mapping generator creates a mapping between two schema versions.

A semi-automatic mapping generator prototype is employed for the XML Schema, which is implemented in Java (Figure 5). The prototype suggests mapping between two XML Schemas by checking similarities in element (tag) names and tree paths (like path-to-path mapping in [3]). A system user can then modify the suggestions, before the final set of mappings are stored as an XML document (as in Figure 4), represented as a special mapping object in the DRL. An extension of the prototype can be the examination of the elements of data types / domains, or the

[1] A project in association with the foundation DigiMed [5] at NTNU.

inclusion of some kind of semantic knowledge, such as, thesaurus (meta-language) (e.g. [1]) into the program for finding potential mappings. So far, the prototype only deals with the syntactic level of differences.

5. Conclusion

More work is required to analyze a common database solution for a future total integrated EHR (Electronic Health Record). The work must argue how a good database and related data representation solution can influence the reality of such a future system, including the data security aspect and the overall system functionality. Further analyzes are needed providing some realistic cases of possible use, such as the case of electronic referral. For the DRL we can implement a test tool that uses mappings both forward and backward in the temporal space when generating replys on queries from client programs.

This presentation gives some headlines for an ongoing PhD work. The background and inspiration to look at a long-term temporal data representation of an EHR are presented. Linked to this is an examination of a possible temporal realization and a mapping between schema versions in a temporal space. A temporal data model is shortly described.

References

[1] van Bemmel JH, Musen MA (ed.). Handbook of Medical Informatics, Springer 1997, ISBN 3-540-63351-0.
[2] http://www.centc251.org
[3] Cluet S., Veltri P., Vodislav D. Views in a Large Scale XML Repository, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.
[4] Dick R.S, Steen E.B, Detmer D.E (Eds.). The Computer-Based Patient Record - An Essential Technology for Health Care, Revised ed., Institute Of Medicine, National Academy Press 1997.
[5] http://www.digimed.no (in Norwegian)
[6] http://www.mannvernd.is/english
[7] Kalinichenko L.A. Integration of Heterogeneous Semistructured Data Models in the Canonical one, http://citeseer.nj.nec.com/kalinichenko99integration.html
[8] http://www.kith.no (in Norwegian)
[9] Lærum, Hallvard. EPJ ved norske sykehus, Presentation at: http://kvalis.ntnu.no/PublicDocs (in Norwegian).
[10] http://www.medcom2.dk/english
[11] Nørvåg, K. VAGABOND The Design and Analysis of a Temporal Object Database Management System, Dr. ing. thesis, Norwegian University of Science and Technology, ISBN 82-7984-097-4
[12] Reynaud C., Sirot J-P, Vodislav D. Semantic Integration of XML Heterogeneous Data Sources. IDEAS. Proceedings, IEEE Computer Society 2001, ISBN 0-7695-1140-6
[13] Fagan L.M, Shortliffe E.H, The Future of Computer Applications in Health Care, Chapter 20 in Medical Informatics - Computer Applications In Health Care and Biomedicine, Springer-Verlag 2001, ISBN 0-387-98472-0
[14] Snodgrass R. Temporal Databases, in Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Springer-Verlag, LNCS 639, 1992

[15] Waegemann C.P. The five levels of electronic health records, in M.D. Computing, Vol.13 No.3 1996

[16] Hakimpour F, Geppert A. Ontologies: an Approach to Resolve Semantic Heterogeneity in Databases, http://www.ifi.unizh.ch/dbtg/Projects/MIGI/publication/ontoreport.pdf

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004. VOLUME 673, Databases and Information Systems
164 174 P.

MQTL – Model Query and Transformation language

Raimonds Praude

University of Latvia, IMCS
29 Raina boulevard, Riga, Latvia
raimonds.praude@sets.lv

Abstract. MDA approach becomes more and more popular in the modeling world. A new Model Query and Transformation language for MDA tools is proposed in this paper. The language is based mainly in OCL, but also contains some ML features and uses the functional approach for queries and transformations

Keywords. Model transformations, MDA, functional language

Introduction

The Model-Driven Architecture (MDA) is an initiative by the Object Management Group (OMG) to define an approach to software development based on modeling and automated mapping of models to implementations. One of the key points of MDA approach is model query and transformations ability. Models in this approach are built in accordance with the corresponding *meta models*, which in turn comply to MOF standards, i.e., meta models are UML class diagrams in a restricted syntax. Models formally are instance sets of the corresponding meta model. Therefore both queries and transformations must be able to process instance sets of a class diagram (meta model). For transformations typically two meta models are involved – the source and the target one.

A *query* is an expression that is evaluated over a model. The result of a query is one or more instances of types defined in the source model, or defined by the query language [1].

Transformation A transformation generates a target model from a source model [1].

There are many proposals for Query and Transformation languages [1], [2], and all of them regard queries as a necessary part of a transformation, i.e., a query determines when and how a transformation (rule) is applicable to a model (or a set of models) and how the result of the transformation will be built:

1. Adaptive Ltd. (in the following abbreviated with ADAPTIVE)
2. DSTC/IBM (abbreviated with DSTC) [3]
3. Compuware Corporation/Sun Microsystems (SUN) [4]
4. Alcatel/Softcam/TNI-Valiosys/Thales (THALES) [5]
5. Kennedy Carter (KC)

6. TCS, which comprises Artisan Software, Kinetum, Kings College, and the University
   of York (TCS) [6]
7. Codagen Technologies Corporation (CODA)
8. Interactive Objects Software GmbH/Project Technology (IO) [7]

For queries several submissions propose the use of the OCL 2.0 language: IO. SUN, TCS. THALES proposes an extension to OCL called TRL. A query in TRL can return not only elements from the queried model as an answer, but can also return a composite type (e.g., a tuple or collection) or a more complex type defined by some meta model.

OCL [8,10] is a textual specification language, developed by OMG, designed especially for the use in the context of diagrammatic specification languages such as the UML, including the use for queries. However, the standard use of OCL for queries in UML is rather limited. Therefore OCL must must be extended, in order to become useful in the more broad context of MDA queries – even the authors of OCL 2.0 admit this [10].

The paper proposes one such extension – MQTL.

MQTL means Model Query and Transformation language. MQTL is based on OCL 2.0, because OCL is a standard in the modeling practice and contains many useful constructions for the new language. However, some features of MQTL come from ML, to extend the expressive power of the language. ML is a widely used functional language [9]. ML has been selected due to the fact OCL is also a functional language in a sense and the basic ideologies of these to languages are compatible. Some of the existing OCL features (*contexts*, for example) can no more be used in this new context and must be modified or deleted as irrelevant. For example, queries of MQTL also can return a composite type (e.g., a tuple or collection) or more complex type defined by some meta model – similarly to the ones in TRL [5]. In general, the goal for building MQTL has been:

- to preserve as much constructs as possible in this extended context
- to include the minimum number of constructs from ML, in order to ensure the required query functionality.

Typical examples of queries in MQTL are the following.

Suppose, that our meta model corresponds to abstract syntax of a programming language, where there are such constructs as Program and Function. A model, specified by this meta model, can, for instance, be used as a cross-reference model for a concrete program system. So, the following query is typical in this case: to find out if a Program object instance transitively calls a Function object instance.

In addition it may be required to optimize somehow this model, reducing the number of Functions, for instance.

A meta model can also specify Local Area Networks (LAN) with the corresponding elements – hub, computer, cable, etc. As the models are represent concrete LANs, we can query if a computer C1 connected to a hub H1 via a Cable CA1 in the given LAN.

A query can include some nontrivial algorithm as well, when the required result is not a simple reformatting of the required part of the source model, but some information to be derived in a complicated way.

MQTL can also be used for model transformations, especially if some nontrivial algorithms must be used or additional static structures for intermediate data must be defined. This the case where the MDA languages based on simple graphical patterns and rules (such as [6]) fail and where the functional power provided by ML becomes essential. The transformation rules in fact are functions, so, MQTL uses functional approach also for the transformations. However, this paper mainly is focused on MQTL as the language for querying purposes. Transformation programs in MQTL require some further research.

The proposed language is more expressive in use than the few existing extensions of OCL for queries, for example TRL [5].

The paper gives a precise description, which of the OCL constructs are preserved in MQTL and what is added from ML. Then an example of nontrivial query in MQTL is provided.

Why does the pure OCL 2.0 is not suited

OCL can be used for a number of different purposes [8]:

- As a query language
- To specify invariants on classes and types in the class model
- To specify type invariant for Stereotypes
- To describe pre- and post conditions on Operations and Methods
- To describe Guards
- To specify target (sets) for messages and actions
- To specify constraints on operations
- To specify derivation rules for attributes
- For any expression over a UML model

However, MQTL can take from OCL only these features and constructs, which are useful for querying ("as a query language").

Each OCL expression must be written in the *context* of an instance of a specific type [8]. In an OCL expression, the reserved word *self* is used to refer to the contextual instance. For instance, if the context is *Person*, then *self* refers to an instance of *Person*.

```
context Person
def: nickname : String = self.name
```

The keyword *self* itself is optional in OCL, but there are no ways to extend the semantics imposed by it, if we don't drop the concept of OCL context completely.

The queries in MQTL are written only within *functions* – the main structural component of MQTL. In fact, functions can be defined also in OCL, they are called additional operations there and are defined via the "*def:*" keyword. However, these

additional operations can be defined only within the *contexts* in OCL. Thus, if the operation A1 is defined in the context B1, it cannot be called from the context B2. It means that according to rules of OCL we cannot reuse a query, written in the operation A1, which is defined in context B1, in the operation A2, defined in the context B2. Therefore, the *context* considerably restricts OCL usage and that's why it is not used in MQTL.

In addition, some of OCL expressions must be modified, because the *context* is not longer used and the expressions are extended with ML features to increase the expressive power of them. The following OCL definitions, expressions, data types are modified:

1. additional attributes and operations definition
2. let expression
3. Sequence data type

The next section provides the list of all OCL constructs which are preserved (or slightly extended) in MQTL. It is followed by the list of constructs included from ML.

MQTL language description

The features and constructs, coming from OCL without modifications

1. **if_then_else** – is the same in MQTL as in OCL.

2. **Tuple in OCL.** It is possible to compose several values into a *tuple*. A tuple consists of named parts, each of which can have a distinct type. Also, the values of the parts may be given by arbitrary OCL expressions. Example:
   `Tuple {name: String = 'John', age: Integer = 10}`
   Tuple {age = 10, name = 'John'} – the type names are optional and the order of the parts is unimportant.

   Tuple in MQTL. The tuple concept is the same in MQTL as in OCL – it also may consist of named parts, each of which can have a distinct type, for instance:

   ```
   val t : Tuple{i : Integer,s : String} = Tuple{i = 10,s = getTitle( )}
   ```

3. **Navigation via associations** is the same in MQTL as in OCL.

4. **Concept of Model data type and Model data type functions, allInstance feature of any Model data type.** The same in MQTL as in OCL.

5. **Collections data types in OCL.** Curly brackets surround the elements of the collection, elements in the collection are written within, separated by commas. The type of the collection is written before the curly brackets: There are four collection data types in OCL: *Set*, *OrderedSet*, *Bag* and *Sequence*. Examples:

```
Set { 1 , 2 , 5 , 88 }
OrderedSet { "1" , "22" }
Sequence { 'ape', 'nut' }
Bag {1 , 3 , 4, 3, 5 }
```

Collection data types in MQTL. Are the same in MQTL as in OCL.

9. **Primitive data types** are the same in MQTL as in OCL.

The features and constructions, coming from OCL with modifications

1. **Operation definition in OCL.** These constructions enable reuse of variables/operations over multiple OCL expressions. All operations are specified via «definition», and are known in the same context as they are defined. For example:

```
context Person
def: getTitle( ) : String = "Person",
```

where *hasTitle*( ) is an operation and *t : String* – the parameter. After '=' symbol in the operation definition can be any OCL expression (let, if_then_else, ...). Recursive operations are also allowed. After *hasTitle( )* operation is defined, it can be used in other OCL expressions within the current *context*.

Operation definition in MQTL.

As *context* is not used in MQTL, the operation becomes the basic construction in the new language and all MQTL expressions are written within operations. So, operation corresponds to a in fact function and that's why it is defined, using *fun* keyword:

```
fun getTitle( ): String = "Person"
```

2. **Attributes (variables) definition in OCL.** These constructions enable reuse of variables over multiple OCL expressions. All variables specified via «definition», known in the same context as they are defined.

```
context Person
def: nickname : String = 'Little Red Rooster',
```

where *nickname* is an attribute. After *nickname* attribute is defined, it can be used in other OCL expressions within the current *context*.

Attributes (variables) definition in MQTL.

As *context* is not used in MQTL, the values can be defined only with operation (function). So, variables are defined not with the keyword *"def:"* but with *val* (it comes from ML), because *"def:"* keyword meant for operations and variables that they are not owned by *context*-class, but are defined as the helper attributes/operations. *val* means that simply a new value (variable) is defined.

```
val nickname: String = 'Little Red Rooster'
```

After '=' symbol can be any MQTL expression.

3. **let expression in OCL.** *let* expression allows one to define a variable, which can be used in the constraint. A let expression may be included in any kind of OCL expression. It is only known within this specific expression. Example:

```
context Person
def: inc(i : Integer, k : Integer) : Integer =
     let
         income : Integer = k
       in
         if i < 10 then
             income + 100
         else
             income + 1000
         endif
```

let expression in MQTL. The constraint that only one variable declaration is allowed within *let* expression requires nested let constructs if more than one variable should be declared inside *let*. So, one or more variable declarations are allowed in MQTL inside *let*:

```
fun mult(i : Integer, j : Integer) =
 let
     val mult1 : Integer = mt(i)
     val mult2 : Integer = mt(k)
 in
     mult1 * mult2
```

The features and constructs coming from ML

1. **The ability to define a new type, record in fact.** This definition is useful if we want to use a type, which is not predefined, in many statements. Example:

```
type p_info = {
   name : String,
   surname : String,
   age : Integer}.

val per1 : p_info = {name = "John", surname =
"Smith"}
```

```
val per2 : p_info = {name = per1.name, surname =
per1.surname}
```

The new type *info* is defined and then in is used twice – in *per1* and *per2* definitions.

2. **local declaration.** Limits the scope of one declaration to another declaration:

```
local
   fun helper (i: Integer ): Integer = i + 1
  in
     fun f1() = helper(2)
  end
```

Here *helper* function can be called only within this *local* declaration, but the function *f1* – from any place of a program. This declaration can be used to create query blocks – on query can be called from the outside, but the other queries in this block are the helpers for this query.

3. **list patterns.** The list patterns:

```
h::t, nil
```

where *h* is the header and *t* – the tail of the list and *nil* – the empty list. This pattern is applied to Sequence data type in MQTL, that's why *h* is a header of the Sequence and *t* the tail of the Sequence, and *nil* – empty Sequence. So, the Sequence has the form: $h_1::h_2::...::h_n::nil$. Such a pattern is used in many functional languages and it significantly extends the expressive power of the new language. The example, given in the next chapter, demonstrates the usage of this pattern.

4. **case expression.** Case expression has the following form:

```
case exp
of pat1 => exp1
| ...
| patn => expn
```

, where pat_1, .. pat_n are the patterns. These patterns are the constants of primitive data types (0, "a"), or the list patterns.

MQTL example

Suppose we have such a meta model:

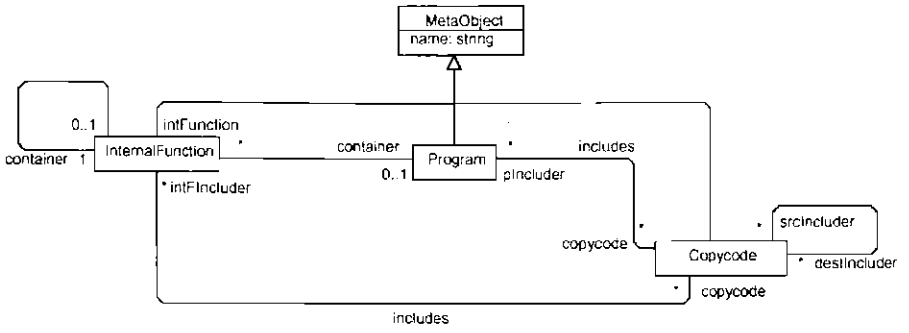


Figure 1

This meta model corresponds to a small part of a programming language and specifies the abstract syntax of programs, which can be written in this language.

Copycode resembles a macro definition in "C++" language. in fact, but it is defined in a separate file. However, all the Copycode definitions, included in a Program (via include operator), are inserted into a program during the compilation process – the same as macro in C++. Also, a Copycode can include another Copycode and so on, but anyway, they are inserted into a Program. which includes the first Copycode in this chain.

Internal functions are defined in the same file as the whole Program. They also can include Copycodes.

Here is the instance diagram of the meta model in Figure 1:

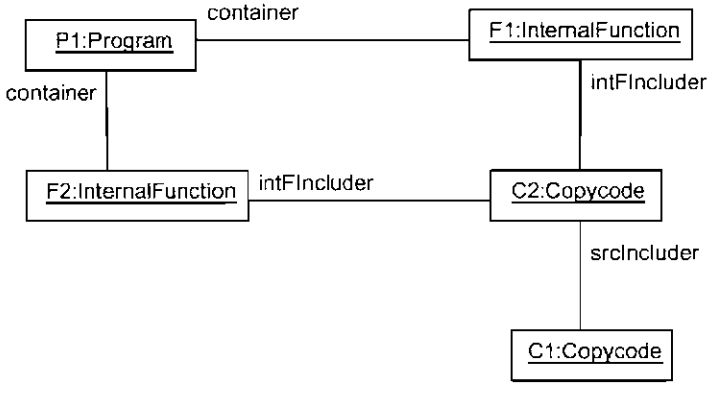


Figure 2

So, Program P1 contains the two InternalFunction instances – F1 and F2. Internal functions F1 and F2 both includes Copycode C2 and Copycode C2 includes Copycode C1.

Suppose we created a cross-reference model, specified by the meta model in Figure 1 and we need to check if Program P1 transitively includes Copycode C1. Transitively includes means, that if, for example, a Program P contains an

InternalFunction F and this function includes a Copycode C, a Program includes Copycode C. The same is for Copycode, including another Copycode. So, **Program P1 transitively includes Copycode C1 in Figure 2.**

It means we should write a query in fact for cross reference check, with the two parameters – P1, C1: **`checkCall(P1, C1)`**. The query should return a Boolean type - True for the given example.

The example code and description

```
fun heckCall(r: equence(ModelElement),cl : ModelElement)
: Boolean = helpCheckCall(r, cl, nil)

fun      helpCheckCall(r:      Sequence(ModelElement),cl:
ModelElement,visited: Sequence(ModelElement)) : Boolean
=
      case r
         of h::t =>
                if (h.name = cl.name) then
                   true
                else
                   if (visited.includes(h))
                      checkCall(t, cl, visited)
                   else
                    checkCall(nextVertexes(h).union(t),
                    cl, h::visited)
                   endif
                endif
         of nil =>
              false

fun nextVertexes(x:ModelElement): Sequence(ModelElement)
=
     let
        val r : Sequence(ModelElement) =
            if (x.IsOfType(Program))
                  x.container.asSequence().union(
            x.pIncluder.asSequence())
            else
               if (x.IsOfType(InternalFunction)
                    x.container.asSequence().union(
x.intFIncluder.asSequence())
               else
                if (x.IsOfType(Copycode))
                     x.srcIncluder
                  endif
               endif
          endif
```

```
    in
      r
    end

val b : Boolean = checkCall(P1, C1)
```

We should traverse the model (graph in fact), starting from P1 node and if we meet a node C1 during the traversal, it means, that P1 includes C1, otherwise – not. The traversal algorithm is depth first search.

Depth first search involves chasing down edges (from some starting nodes) to collect up all nodes which are reachable from those start nodes. The exploration should go down one chain from a start node, returning these nodes before backing up to look at the next deepest unexplored nodes. Clearly we need to keep track of which nodes have been seen before so that we do not go round in circles whenever there is a cycle in the graph. The central function (*helpcheckCall*) has a list of nodes (to be explored nodes), a node to be checked for inclusion and a list of 'visited' nodes as parameters.

nextVertexes function returns the list of the nodes, connected to x.

Conclusions

The paper describes the basic principles of the query and transformation formation language MQTL. The main innovative element of the language is ML constructs, increasing the expressive power of MQTL. The given example shows only a small part of the constructs and power of MQTL, however it demonstrates the functional approach, which is the base of MQTL, and its natural recursive style for computing a transitive closure.

References

[1] Meta modelling for MDA First International Workshop York, UK, November 2003 Proceedings

[2] Krzysztof Czarnecki and Simon Helsen, Classification of Model Transformation Approaches, University of Waterloo, Canada

[3] OMG: MOF Query / Views / Transformations, http://www.omg.org/docs/ad/03-08-03

[4] OMG: XMOF Queries, Views and Transformations on Models using MOF, OCL and Patterns, http://www.omg.org/docs/ad/03-08-07

[5] OMG: Response to the MOF 2.0 Query / Views / Transformations, http://www.omg.org/docs/ad/03-08-05

[6] OMG: Revised submission for MOF 2.0 Query / Views / Transformations RFP, QVT-Partners, http://www.omg.org/docs/ad/03-08-08

[7] OMG: Revised Submission to MOF Query / View / Transformation RFP, http://www.omg.org/docs/ad/03-08-11

[8] OMG Response to the UML 2.0 OCL RfP, http://www.omg.org/docs/ad/02-05-09

[9] Programming in Standard ML, Robert Harper, Carnegie Mellon University Spring Semester, 2001

[10] The Object Constraint Language 2nd ed, J. Warmer, A. Kleppe, Addison Wesley, August 29. 2003

[11] MDA Explained The Model Driven Architecture: Practice and Promise, A. Kleppe, , J. Warmer, W. Bast, 2003, Addison Wesley, 2003

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
175–185 P.

Publishing Relational Data to Construct Recursive XML Documents

Lavr Burin

University of St Petersburg, Russia
lburin@yandex.ru

Abstract. The problem of publishing relational data as XML documents is considered in the case of presence of recursion in XML schema view. While XML is a standard convenience for exchanging business data on the World Wide Web, relational database systems represent the most convenient way for storing large volumes of data; hence finding an efficient mechanism for publishing relational data over XML becomes a highly important task. Analysis of real-life situations shows that the presence of recursion in XML documents is quite normal. At the same time there is an evident lack of issues in the area. It seems particularly promising to combine the support of recursive XML documents with the current support of linear recursion in SQL99 using special language constructions. A technique for constructing recursive queries to extract XML data from relational database by means of SQL is proposed in this paper. Based on relational data with some additional information XML view over this data can be easily materialized.

1. Introduction

Over the recent years XML has emerged as the standard for exchanging business data on the World Wide Web. Its structure provides a series of advantages for applications in data modeling and exchanging. At the same time, however, most of business data is stored in relational database systems and the situation here will remain so for the foreseeable future, because of the scalability, reliability and performance associated with relational database systems. Thus, if XML is to fulfill its potential, finding efficient ways of publishing relational data as XML documents becomes a particularly important problem. XML publishing is one of the most challenging problems [15] and it is the problem of transforming existing relational data into XML. Conceptually, this is the same as defining an XML view over the relational data.

The definition of XML view consists of two main requirements: the *view definition language* and the *XML schema of the resulting view*. The language specification describes how to structure and tag data from one or more tables as a hierarchical XML document. The structure of XML data, by contrast to relational data, is nested and self-describing. Usually XML documents conform to their input DTD. XML Document Type Definitions (DTDs) [18] describe the structure of XML documents and are considered as the schemas for XML documents. More

sophisticated tools for structuring XML such as XML-Schema [2, 16] are based on DTDs.

The problem of XML publishing becomes much more interesting and difficult in case of the presence of recursion in XML view schema. XML schema is said to be recursive if it has some element type defined in terms of itself, directly or indirectly. As for real-world XML schemas, a number of DTDs are analyzed in [4] and the result is that more than half of them are recursive; thereby in practice the presence of recursion in XML schema is quite usual. As noticed in [1], recursive DTDs are commonly found in specifications of biomedical, chemical and protein data. As example, Figure 1 shows a fragment of schema graph taken from BIOML [17] content model. This model corresponds to real-world logics: *DNA* is specified in terms of *clone*, *clone* has sub-elements *DNA* and *gene*, while *gene* is in turn specified with *DNA* etc.

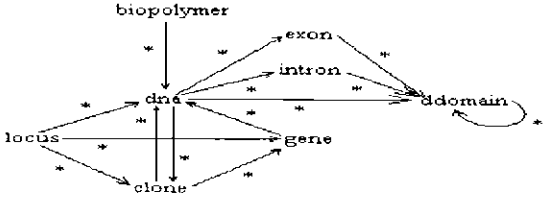


Figure 1. Sample XML schema.

Studying DTDs from different application areas we can note that there are a lot of XML schemas with recursion involved in XML publishing. On the other hand, as shown in Section 2, there is an obvious lack of solutions to the problem of recursion support in XML publishing area. Thus, since support of recursive XML schemas seems important, it is useful to propose some mechanism for transforming relational data into XML with respect to the presence of recursion in predefined XML schema of the resulting view.

Information is recursive itself in numerous areas, connecting both science and industry. Entities defined in terms of themselves, explicitly or implicitly, occur particularly in biomedicine, biology, chemistry, molecular theory, electronics, programming, geography as well as in different classifications of complex systems, in part hierarchies etc. Significant effectiveness and accuracy might be achieved, for example, in storage of large volumes of information about molecular structures (e.g. DNA) in relational databases. We could use all the power of contemporary industrial RDBMSs, but representing of the information will be done in standard and comprehensible XML. Also, research studying in [12] outlines that it is far more efficient to construct XML document inside the relational engine than doing so outside.

Imagine that we have some XML data (for example, we have any XML representation of original text data), but this XML representation doesn't reflect the natural structure of our data, thus we wish to obtain efficient and, at the same time, natural data representation over XML, specifically regarding the support of recursion. As a first step, let us store that XML data in a relational database. The main goal of storing data in relational database is to put this data into an efficient storage schema, which is likely to include special metadata needed for the support of

recursion (one or more columns of relational table). Then, using the technique of recursive queries, we will obtain XML views of that relational data and these XML views will be recursive. Thus, we will solve the *XML publishing problem* for complex data, at the same time presenting it in its *natural form*, i.e. supporting the recursive character of this data.

The *SQL99* standard has the support for linear recursion using *with* clause, so it gives an idea to use this language construction within SQL queries to publish XML data. The proposed technique is mainly based on graph analysis of XML schema of the resulting view. Depending on what edges in XML schema graph we discovered, we construct corresponding SQL query in both cases using *with* clause. The most interesting situation is when we deal with the recursion in a graph. Then we have to define some recursive relation, which actually defined through itself. Some assumptions about transformations between relational data and XML also were done and briefly discussed. As an example, there is the SQL query given in Appendix retrieving data from relational tables in order to construct XML document, conformed to XML schema view which graph is shown on Figure 1. There are numerous unions of different sub-queries to be performed to extract necessary data.

The rest of this paper is organized as follows. In Section 2 we discuss related work. In Section 3 the data model is considered. The technique of using SQL support for recursion to construct recursive XML documents over relational data proposed in Section 4. Conclusions and ideas for future work are presented in Section 5.

2. Related Work

The problem of XML publishing is very attractive to a lot of researchers, because of the aspects mentioned above, thus there are a lot of works in this area. A number of approaches have been proposed on translating XML queries into SQL. In ROLEX [3], SilkRoute [5, 6] and XPERANTO [10, 11, 12, 13] XML views over the relational data are defined using the view definition languages. The focus of these XML publishing systems is on non-recursive XML view schemas.

In [13] the technique for querying XML documents using a relational database system is proposed. Reconstruction of XML view mentioned in this work is generated by creating so-called *default XML view*, which is low-level XML view of the underlying relational database. Definition of other XML views is possible on top of the default view using XQuery language [22].

In [12] SQL language extension to specify the construction of XML documents is proposed. Although some of the techniques proposed show a significant performance benefit, support of the unbound hierarchies is not considered in this work.

Major industrial DBMS vendors also deal with the creating of XML views. Microsoft SQL Server 2000 SQLXML [9, 20] and Oracle XML DB [21] use annotated XSD XML schema defining the XML view, and IBM DB2 XML Extender [19] uses a Document Access Definition (DAD) files. In fact, recursive XML views are supported only in XML DB, while in SQLXML there is support

only for limited number of depths of recursion. XML Extender facilities provide only non-recursive XML views.

The problem of XML publishing in respect to a fixed DTD to which the output documents conform is also considered in [1], where authors refer to this problem as *DTD-Directed Publishing*. This is one of a few works where recursive XML view schemas are allowed, but there SQL (SQL99) support for recursion is not used and all the recursion is implemented in middleware. The notion of *attribute translation grammars* (ATGs), which proposed in this paper, is the mechanism of DTD extending by associating semantic rules via SQL queries. However, this technique has nothing with the support for recursion currently existed in SQL99 [7].

3. Data Model

3.1. On DTD

XML DTDs [18] describe the structure of XML documents and are considered as the schemas for XML documents. For the sake of perception, our data model differs from original DTD data model with unessential differences, discussed below.

We model both XML elements and XML attributes as XML elements since XML attributes can be considered as XML elements without further nesting structure.

We ignore all encoding mechanisms used in data types *CDATA* and *PCDATA*, modeling all of them as *string* data type.

A DTD D is modeled as the set of XML element definitions $\{d_1, d_2, \ldots, d_k\}$, where each XML element definition d_i is in the form of $n_i = e_i$, where n_i is the name of an XML element, and e_i is the DTD expression. Each DTD expression is composed from XML element names which are, in fact, the most primitive DTD expressions, and other DTD sub-expressions using the standard operators from regular expressions theory. Thereby, we can define DTD expression in BNF notation as follows:

$$e ::= string \mid name \mid e \mid e? \mid e^* \mid e+ \mid (e_1 \mid \ldots \mid e_n) \mid (e_1, \ldots, e_n),$$

where *name* is from XML element names and $e, e_1, \ldots, e_n$ are DTD expressions.

Within the DTD model using the *DOCTYPE* declaration defines which element is used as the schema for XML documents. We call this element as the *root* element.

It is obvious that our data model, mainly based on DTD as it described above, entails the graph representation as the directed graph $G = (V, E, root)$, where V is the set of vertices and E is the set of edges, $root \in V$. The vertices represent elements within XML schema and are labeled with their names while the edges represent parent-child relationships between elements and have the multiplicity label from one of ?, *, −. If there is no multiplicity label associated with the edge, it means that we have a single occurrence of the destination DTD sub-expression within the source DTD sub-expression. Due to evidence and simplicity we omit an exact formal describing of how to represent the given DTD in form of directed graph.

3.2. On Relational Data and XML Schema Graph

Studying numerous research efforts related to XML storage problem such those mentioned in [14], as well as technique proposed in [13], we can describe correlation between relational data and DTD elements in XML schema during transforming of relational data into XML. Thus we can see that elements correspond to either relational table or some column of the relational table. There is also a possible situation when there is no relational table or column of the relational table to correspond to some element during publishing. Also we have to realize which relationships of the underlying relational schema were transformed into parent-child relationships in XML view schema represented as the edges of XML schema graph. Intuitively they should be projection, selection and join conditions where tables and columns corresponded to the nodes are involved.

As for relational data, we assume that each relational table has an *ID* column as its primary key. Also, probably some of the relational tables have a *parentID* column, which being a foreign key to some other table, is needed to reconstruct parent-child relationship in XML view. In such cases, when some element has more than one parent element, additional column like *parentType* is needed to handle the information for reconstructing parent-child relationship.

The goal of this brief argument is to show what kind of underlying relational data we should expect if we talk about publishing this data into *recursive* documents. The reason is that for reconstructing recursive XML documents we certainly need to deal with parent-child relationships and a situation of multiple parents.

3.3. On Edges Classification in XML Schema Graph

Let n be the non-root node in XML view schema graph. Consider the incoming edge e. Let do back tracking by this edge to its source, say, node m and then do the same thing for all edges incoming to m etc. If on some step of this procedure we reach the node n again, it means that there is a cycle in XML schema graph. So, we know now that the edge e is related to cycle, where n is involved as one of the vertices of this cycle. It is evident that we can do such procedure for each incoming to n edge. Thus we can achieve the partition of the set $I(n)$ of incoming to n edges, namely $I_{cyclic}(n)$ which is the set of edges from the cycle to which n belongs and $I_{acyclic}(n)$ which is the set of incoming to n edges not involved into the cycle with n presented as its vertex.

4. SQL Query View Definition Constructing

Depending on the type of edges in XML schema of the resulting view graph different SQL queries must be executed to retrieve data needed to construct XML view according to existed XML view schema. Here we consider two different situations, occurring in respect of the partition of the set of incoming edges, referred to above.

4.1. Edges not Involved into Recursion Processing Directly

Here we consider the edges from the $I_{acyclic}(n)$ set, where n is the given node. Let $I_{acyclic}(n) = \{e_1, \ldots, e_k\}$ and $n_1, \ldots, n_k$ are the sources of $e_1, \ldots, e_k$ correspondently.

Assume that with each node n_i relational table T_i is associated. Assume also only two possibilities related to node n: either relational table T associated with n or some column A of the relational table T associated with n (although this assumption is quite limiting, it is still a broad area of interest when studying recursive XML schemas). Then the following query retrieves relational data needed to construct XML view from the underlying relational schema.

```
Q':  with T' as (
          select T.*
          from T1, T
          where T1.id = T.parentid and
                T.parenttype = T1.type
       union all
     ...
       union all
          select T.*
          from Tk, T
          where Tk.id = T.parentid and
                T.parenttype = Tk.type
     ) select T'.* (or select T'.A)
```

4.2. Edges Directly Involved into Recursion Processing

In this section we consider node n such that $I_{cyclic}(n) \neq \varnothing$.

Let edges $\{e_{ij}\}_{ij}$ form cyclic sub-graph C within XML schema graph, where e_{ij} is the edge from node n_i to node n_j. Let C is the greatest cyclic sub-graph of given XML schema graph containing n. It means that adding of some new node to C from the remainder vertices (with all correspondent to it incoming or (and) outgoing edges) will change this graph so, that either new vertex will be unreachable from n, or n will be unreachable from this vertex.

Let nodes $n_1, \ldots, n_k$ are the nodes of C such as they have incoming edges and these edges do not belong to C. Regarding the previous section, we may generate for each node n_i query Q_i' which defines relation T_i' (we again assume the same things about the sources of all of incoming edges). Let with nodes $\{n_j\}_j$ which are not from $\{n_1, \ldots, n_k\}$ set, relational tables $\{T_j'\}_j$ are associated. Let all the tables $\{T_i'\}_i$ have the same set of attributes, otherwise we always can add new attributes to existed tables to reduce tables to the unified form. We assume here that all these relational tables have an *id* field as a primary key and a *parentid* field as a foreign key. Also the originality of number i for all the nodes of sub-graph is provided, so we can use i as the identifier for relational table T_i'.

If we now wish to extract all the data from relational tables needed to construct the fragment of recursive XML document according to schema, represented by the C as mentioned above, we can compose the following Datalog [8] query.

```
D'':  T'' ← T1', ... , T'' ← Tk',
      T'' ← T''(id,_,_) & Tj'(_,id,_), ∀j
```

Using SQL construction for defining recursive relations we can construct the following SQL query, defining T'' relation. We define T'' as the relation with the same set of columns as all $\{T_i'\}_i$ relations, defining one additional column called *nodeid* needed to identify the type of original data, being integrated into the complex recursive relation T''. Having *nodeid* field we may build recursive SQL query using such construction of SQL99 standard as *with* clause. Note also, that above we used *with* clause constructing non-recursive queries, because this construction, in general, provides support of temporary relations within single query.

```
Q'':  with recursive T'' as (
          (
            select T1'.*, 1
          union all
            ...
          union all
            select Tk'.*, k
          )
        union all
          (
            select Tj'.*, j
            from T'', Tj'
            where T''.id = Tj'.parentid and
                  T''.nodeid = i and
                  Tj'.parenttype = Ti'.type
          union all
            ...
          )
      ) select T''.*
```

where i and j in Q'' are such that there is the edge from n_i to n_j. The union over all edges in C is taken in Q''.

5. Conclusion and Future Work

A problem of publishing recursive XML documents over relational data was considered and the technique of using recursion support constructions in SQL99 to produce XML views over relational data was presented.

In general, recursion in XML schema presents many issues for further research work. At the very least, all the previously proposed methods in XML publishing area should be extended, while there is support of recursive XML documents.

According to different techniques proposed in [12], we can now compare different approaches to publishing relational data when constructing recursive XML documents. It is outlined in this paper that different approaches have different performance characteristics in different cases, so it is interesting to study what the

situation will be if we have recursive XML data sets. It is not evident that the parameters considered in [12] will remain the same.

Another issue is to study, in a more detailed way, are transformations between relational and XML data to eliminate the limitations which we assumed in this paper. Thus, because of studying of schema nodes processing in case of neither relational table nor the column of relational table is associated with node is not finished completely, it makes this research direction also very interesting. The most important thing is to understand clear the semantics of edges in XML schema graph with respect to relational model of the underlying database.

Other research directions include combination of methods of XML view materialization previously developed for non-recursive XML schemas with approach proposed in this work. In particular, studying the impact of parallelism seems a very challenging but potentially highly rewarding area.

References

[1] M. Benedikt, C. Y. Chan, W. Fan, R. Rastogi, S. Zheng, and A. Zhou. DTD-Directed Publishing with Attribute Translation Grammars. *In Proceedings of VLDB, 2002.*

[2] P.V. Biron and A. Malhotra. XML Schema Part 2: Datatypes. W3C Recommendation. http://www.w3c.org/TR/xmlschema-2, *May 2001.*

[3] P. Bohannon, H. Korth, P.P.S. Narayan, S. Ganguly, and P. Shenoy. Optimizing view queries in ROLEX to support navigable tree results. *In Proceedings of VLDB, 2002*

[4] B. Choi. What Are Real DTDs Like. *In WebDB, 2002.*

[5] M. Fernandez, A. Morishima, D. Suciu, Y. Kadiyska and W.C.Tan. SilkRoute : A Framework for Publishing Relational Data in XML. *IEEE Data Engineering Bulletin, 24(2), 2001 Database Systems, Vol. V, No. N, Month 20YY, Pages 1–55.*

[6] M. Fernandez, D. Suciu, and W.C. Tan. SilkRoute: Trading Between Relations and XML. *In proceedings of 9th WWW, 2000.*

[7] S.J. Finkelstein, N. Mattos, I.S. Mumick, and H. Pirahesh. Expressing Recursive Queries in SQL. *ISO WG3 Report X3H2-96-075, March 1996.*

[8] H. Garcia-Molina, J.D. Ullman, and J. Uidom. Database Systems: The Complete Book. *Pretice Hall, Upper Saddle River, New Jersey 07458.*

[9] M. Rys. Bringing the Internet to Your Database: Using SQL Server 2000 and XML to Build Loosely-Coupled Systems. *In BTW 2001.*

[10] J. Shanmugasundaram, D. Florescu, E. J. Shekita, M. Carey, Z. Ives, Y. Lu, and S. Subramanian. XPERANTO: Publishing Object-Relational Data as XML. *In informal Proceedings of WebDB, 2000.*

[11] J. Shanmugasundaram, J. Kiernan, E. J. Shekita, C. Fan, and J. Funderburk. Querying XML Views of Relational Data. *In Proceedings of VLDB, 2001.*

[12] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pirahesh, and B. Reinwald. Efficiently Publishing Relational Data as XML Documents. *In Proceedings of VLDB, 2000.*

[13] J. Shanmugasundaram, E. Shekita, J. Kiernan, R. Krishnamurthy, S. D. Viglas, J. Naughton, and I. Tatarinov. A General Technique for Querying XML Documents using a Relational Database System. *SIGMOD Record, 30(3), 2001.*

[14] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. DeWitt, and J. Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. *In Proceedings of VLDB, 1999.*

[15] D. Suciu. On Database Theory and XML. *SIGMOD Record 30(3), September 2001.*

[16] H. Thompson, D. Beech, M. Maloney and N. Mendelsohn. XML Schema Part 1: Structures. W3C Recommendation. http://www.w3c.org/TR/xmlschema-1, May 2001.

[17] BIOpolymer Markup Language (BIOML). http://xml.coverpages.org/bioml.html/.

[18] Extensible Markup Language (XML) 1.0 (Second Edition). http://www.w3.org/TR/2000/REC-xml/. 2000.

[19] IBM DB2 XML Extender. http://www-3.ibm.com/software/data/db2/extenders/xmlext/index.html/.

[20] Microsoft SQLXML and XML Mapping Technologies. http://msdn.microsoft.com/sqlxml/default.asp/.

[21] Oracle9i XML Database Developer's Guide - Oracle XML DB Release 2 (9.2).http://otn.oracle.com/tech/xml/xmldb/content.html/.

[22] W3C Working Draft. XQuery 1.0: An XML Query Language. http://www.w3.org/TR/xquery/. 2003.

Appendix. Reconstruction of XML View

In this appendix resulting SQL query is considered. This query extracts from the underlying relational data all the information needed to construct XML view over this data according to the XML schema shown on Figure 1. *With* clause is used to construct recursive queries as well as to hold temporary relations.

```
with recursive DNA as (
    ( select dna.*, "dna"
      from biopolymer, dna
      where biopolymer.id = dna.parentid and
            dna.parenttype = biopolymer.type and
            biopolymer.type = "dna"
    )
  union all
    ( select dna.*, "dna"
      from DNA, dna
      where dna.parentid = DNA.id and
            dna.parenttype = "locus"
            DNA.nodeid = "locus"
  union all
      select dna.*, "dna"
      from DNA, dna
      where dna.parentid = DNA.id and
            dna.parenttype = "gene"
            DNA.nodeid = "gene"
  union all
      select dna.*, "dna"
```

```
      from DNA, dna
      where dna.parentid = DNA.id and
            dna.parenttype = "clone"
            DNA.nodeid = "clone"
  union all
      select gene.*, "gene"
      from DNA, gene
      where gene.parentid = DNA.id and
            gene.parenttype = "clone"
            DNA.nodeid = "clone"
  union all
      select gene.*, "gene"
      from DNA, gene
      where gene.parentid = DNA.id and
            gene.parenttype = "locus"
            DNA.nodeid = "locus"
  union all
      select clone.*, "clone"
      from DNA, clone
      where dna.parentid = DNA.id and
            dna.parenttype = "dna"
            DNA.nodeid = "dna"
  union all
      select clone.*, "clone"
      from DNA, clone
      where clone.parentid = DNA.id and
            clone.parenttype = "locus"
            DNA.nodeid = "locus"
    ) } select * from DNA

  union all

with EXON as (
      select exon.*
      from DNA, exon
      where exon.parentid = DNA.id
), INTRON as (
      select intron.*
      from DNA, intron
      where intron.parentid = DNA.id
), recursive DDOMAIN as (
    ( select ddomain.*
      from ddomain, EXON
      where ddomain.parentid = EXON.id and
            ddomain.parenttype = "exon"
```

```
union all
    select ddomain.*
    from ddomain, INTRON
    where ddomain.parentid = INTRON.id and
          ddomain.parenttype = "intron"
union all
    select ddomain.*
    from ddomain, DNA
    where ddomain.parentid = DNA.id and
          ddomain.parenttype = "dna"
  )
union all
  ( select ddomain.*,"ddomain"
    from DDOMAIN, ddomain
    where DDOMAIN.id = ddomain.parentid and
          DDOMAIN.nodeid = "ddomain" and
          ddomain.parenttype = "ddomain"
  ) ) select * from DDOMAIN
```

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA, 2004. VOLUME 673, Datbases and Information Systems
186–197 P.

Business Process Measures

Valdis Vitolins

University of Latvia, IMCS, 29 Raina blvd, LV-1459, Riga, Latvia
valdis_vitolins@exigengroup.lv

Abstract. The paper proposes a new methodology for defining business process measures and their computation. The approach is based on metamodeling according to MOF. Especially, a metamodel providing precise definitions of typical process measures for UML activity diagram-like notation is proposed, including precise definitions how measures should be aggregated for composite process elements.

Keywords. Business process, model, metamodel, measure.

1. Introduction

Globalization and increasing competition forces companies to improve their business, but improvements can't be evaluated without measurements. Thus, measures of business processes in companies provide crucial knowledge of TCO and ROI for executives. Steady growth of IT usage in business makes these measurements more feasible [1].

Many quality [2,3] and business process management [4,5,6] methodologies now use numeric methods to figure out weaknesses and strengths of a business. Methodologies are supported by several tools [7,8,9,10], however they provide the "best of breed" methodology only for one narrow area and they can't support several methodologies simultaneously.

In this paper the research on business process modeling problems [11] is continued. Measures and rules, they relation to business concepts are shown in a formal and unambiguous way, using Unified Modeling Language (UML) [12]. Theories and methodologies are analyzed using metamodeling approach according to Meta Object Facility (MOF) [13].

Gradually, through given examples for each abstraction (meta) layer, several aspects of business process measures are precisely defined. It is shown that on the one hand, each higher abstraction layer describes concepts that are more common, but on the other hand, each higher layer determines rules and possibilities for lower layers.

The model (M1 layer) is represented by a business process example (UML activity diagram extended by measures) and a class diagram defining the "measure view" of the same example. The most significant and more detailed is metamodel (M2 layer), because it determines common possibilities and features for business process models in a modeling notation. Extended metamodel shows how standard measures and measure aggregation facilities for composite objects are defined. The classification of reasonable process measures and their possible assignment to

process elements is provided. The metametamodel (M3 layer) briefly sketches a universal framework for measure definition, from which the specific "measuring metamodel" (M2) could be obtained as an instance.

The proposed metamodel can be used as a unified framework for the development of comprehensive business modeling and measurement tools.

2. Business Process Model (M1)

To demonstrate measuring of a business process on a practical example, a specific modeling language will be used. This language is a slightly modified UML Activity diagram (AD) [12] with **extensions for object measures**. Mainly the graphical notation of activity diagram is made more expressive and the terminology is changed, but the semantics is a standard one, except the resource management, which is made more precise. The language corresponds also to the previously developed business process metamodel [11]. References to diagram elements are shown in *italic* in the text. The example (Fig. 1) shows one business process for a shop, which delivers pizzas to customer homes. *Sell Pizzas* is a business process (an activity in AD notation), which consists of several tasks (actions in AD).

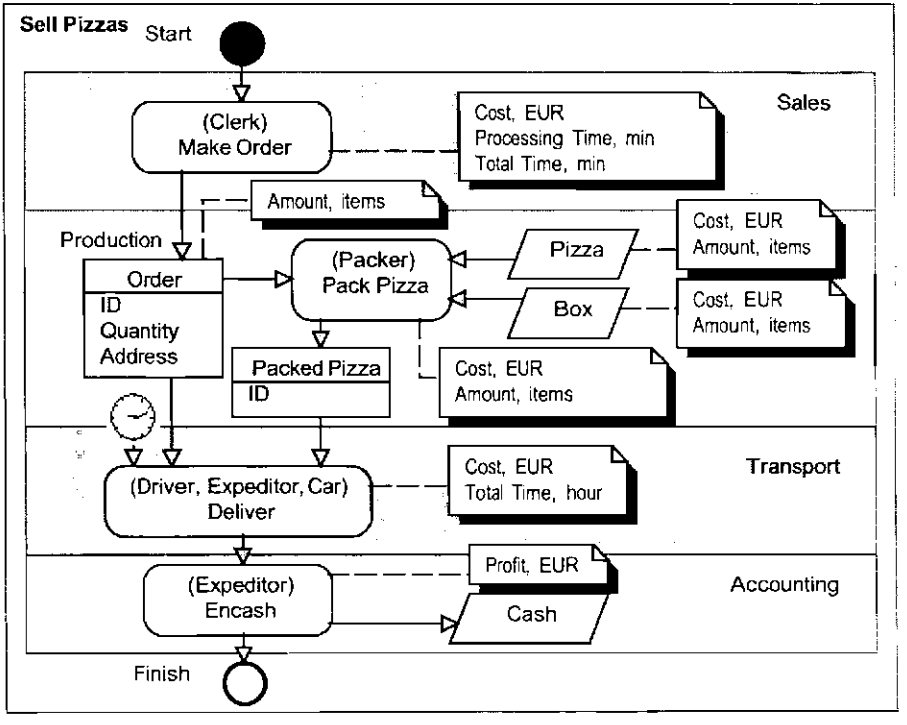


Fig. 1 Sell Pizzas business Process (M1)

Tasks are represented by rounded rectangles. Some tasks have **performers** in parentheses. Performer is a special reference that is used to specify that the given

employee or resource is necessary to perform the task. Control flows are shown by simple arrowed lines, and rectangles represent objects in object flows – either flows of messages (e.g. *Order*) or physical objects (e.g. *Packed Pizza*) with their attributes. Parallelograms correspond to AD datastores, where materials are located. An object flow entering a datastore means that putting into the datastore, but leaving flow - taking. The fact that *Deliver* is a task, which is started at some regular time moments is shown by a symbolic clock (Time event in AD). By default, all incoming flows are joined at a task with AND condition. Organizational units that are responsible for each task (e.g. *Sales*) are shown as swim lanes. If a specific performer for a task isn't set, anyone from the corresponding organizational unit can perform the given task. If performer is set, only the given performer can perform the given task. Each process (or subprocess) starts with *Start* node and finishes with *Finish* node.

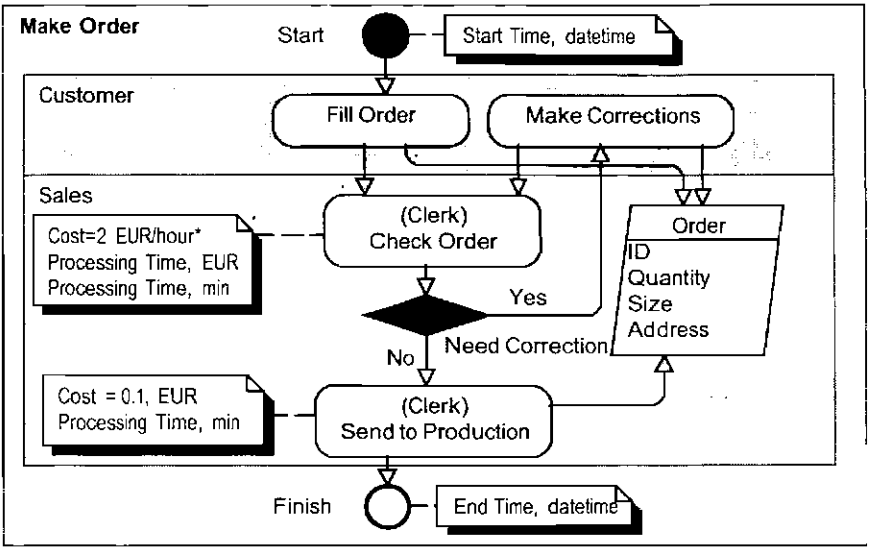


Fig. 2 Make Order business process (M1)

Make Order in Fig. 2 actually is another business process which is decomposes the task *Make Order*. This subprocess contains a decision *Need Correction* (diamond). The complete *Sell Pizzas* business process is shown only for a "full picture" and to demonstrate that several measures for some business process objects can be assigned. **For further explanation, only the *Make Order* subprocess and *Make Order* task will be used.**

Measures are shown in notes (e.g. *Cost*, *EUR*), which are linked to the measured element (object) by a dashed line. Each measure in a note is specified by its name, an optional declaration and the unit according to the following syntax: Name[=declaration],Unit (e.g. *Cost=2 EUR/hour*Processing Time, EUR*). The measure declaration can be empty, a constant or expression. If a measure declaration references a name of other measure, by default the measure for the same object is assumed. From analysis of several process management tools [7, 9, 10], it was assumed that each system provides information about system time and performers. Therefore, measures, which rely directly on system time and performers don't need

declarations. All other measure values should be declared implicitly in metamodel, or explicitly as constants or formulas in model.

A measure linked to a process element means that the given measure must be evaluated during the process execution. See more on measure values in sections 3 and 9. According to MOF this business process model is an abstraction of all execution instances of the real business process therefore it conforms to M1 layer.

For clear understanding of all metalayers, an example of instance layer M0 also will be shown.

3. Measure Value Instances (M0)

During the system run/simulation of the business process, shown in Fig. 1 and Fig. 2, each of the defined objects in the model gets its runtime instance. Each runtime instance of an object has all its attribute values set, including the values of measures linked to the object. In Tab. 1, instances are shown as a "denormalized" table (view), where **rows correspond to the registered measure instances**.

Instances from the *Make Order* sub-process are shown in this table for one process execution. Columns *Object Type* and *Object* represent the owning business object instances. Cells in the *Measure Declaration, Unit* column represent the identification of each measure. All the above-mentioned cells are actually resolved from business process model (they are not instance dependent). Cells in *Value* and *Time* columns represent values for each measure instance. Cells in these columns are filled by the process management system according to the actual process execution. *No, Source No* columns are for information only and describe how derived values are calculated.

The table illustrates that some values are got explicitly from the management system (e.g. value for *Processing Time*), but some values are calculated implicitly through given rate and explicit value (e.g. *Cost* from *2 EUR/hour* and *Processing Time*), or from values of several sub-measures (e.g. *Cost* for *Make Order*, from two separate costs).

No	*Object Type*	*Object*	*Measure Declaration, Unit*	*Value*	*Time*	*Source No*
1	Business Process	Make Order	**Start, datetime**	**9:05:34**	**9:05:34**	-
2	Task	Check Order	**Processing Time, min**	**0:03:00**	**9:10:03**	-
3	Task	Check Order	**Cost=2 EUR/hour*Processing Time, EUR**	**0,10**	**9:10:04**	2
4	Task	Send to Production	**Processing Time, min**	**0:02:00**	**9:15:40**	-
5	Task	Send to Production	**Cost =0.1, EUR**	**0,10**	**9:15:45**	-
6	Business Process	Make Order	**Finish, datetime**	**9:15:47**	**9:15:47**	-
7	Business Process	Make Order	**Processing Time, min**	**0:05:00**	**9:15:47**	2,4
8	Business Process	Make Order	**Total Time, min**	**0:10:13**	**9:15:47**	1,6
9	Business Process	Make Order	**Cost, EUR**	**0,20**	**9:15:47**	3,5
10	Business Process	Make Order	**Processing Time, min**	**0:05:00**	**9:15:48**	2,4

Tab. 1 Sample Values for Make Order Sub-process

General rules according to which this table could be obtained from the "raw material" – a complete process execution log are described in section 9.

4. Measure Aggregation Sample Model (M1)

Though the activity diagrams in Fig. 1 and Fig. 2 represent the M1 layer, they are only "graphic interfaces" for the control aspects of the business process model, and in such representation not all measure-related items are viewable. Therefore, the measure aspect of the same model is shown in full details, using a class diagram (Fig. 3). This class diagram is another view for the same process, where "system" objects that actually exist and are necessary for process measuring are made explicit. At the same time, the control and execution aspects of the model are not visible in this view.

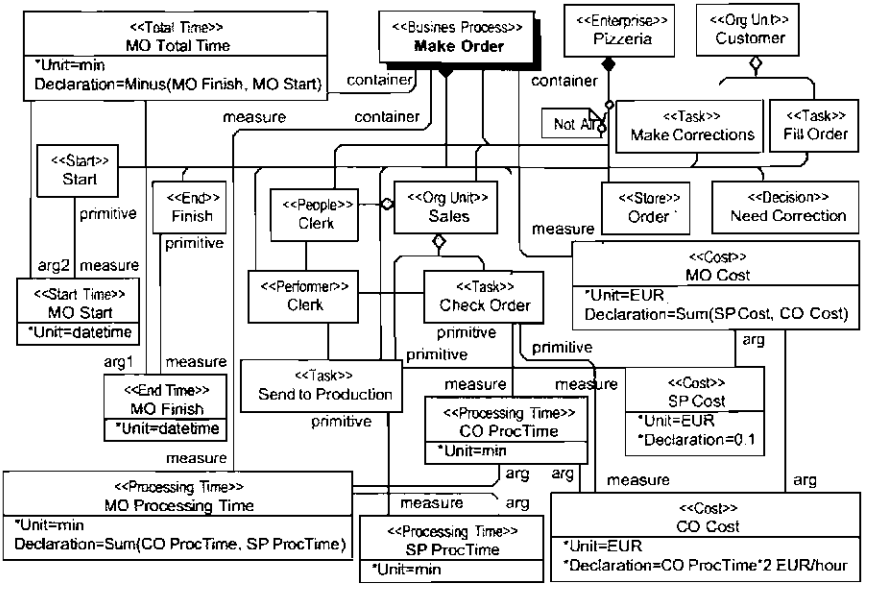


Fig. 3 Measure Sample Model (M1)

The diagram is built in accordance with the process measuring metamodel (M2 level) in Fig. 5. E.g., object relations to their measures are shown as relations with the specified role names *primitive/container-measure*. If a measure declaration references another measure, it is shown with role name *arg*.

In this diagram on the one hand each class is one specific **instance** of a more abstract class in M2 layer (Fig. 5), but on the other hand it is a **class**, because it represents all possible instances from the M0 metalayer (Tab. 1). So this means, that they all are classes with more specific properties than classes in M2 metalayer. Let name such classes "instance classes" (it is an extension of UML class diagram notation). This is shown by using stereotypes in such a way, that a class name in a higher metalevel (e.g. M2) becomes to the stereotype for the corresponding instance classes in the lower metalevel (e.g. M1).

According to the metamodeling traditions in MOF (including the metamodel for UML class), components that compose a measure at M2 metalayer, are shown as new specific tagged compartments of a concrete measure class at M1 layer (e.g.

unit=EUR, declaration=Minus(MO_Finish,_MO_Start)). Compartments, what are explicitly defined in business process model (Fig. 1, Fig. 2), are marked with asterisk. Unmarked compartments are derived implicitly from measure declaration metamodel.

Unfortunately, a universal application of this principle doesn't work well always. E.g., if this rule would be used for *Business Process* class, it should be shown as a single class with compartments at M1 layer. In such way, diagram would become too unreadable, therefore, some decompositions at M2 layer are treated as decompositions at M1 layer. E.g., *Business Process* and *Enterprise* are shown as usual decomposition with separate classes and corresponding stereotypes.

5. Business Process Metamodel (M2)

In Fig. 1 a model of one particular business process (M1) was shown, but for the development of a universal process measuring method, an adequate business process metamodel (Fig. 4) is necessary (M2 layer), as the base for that notation.

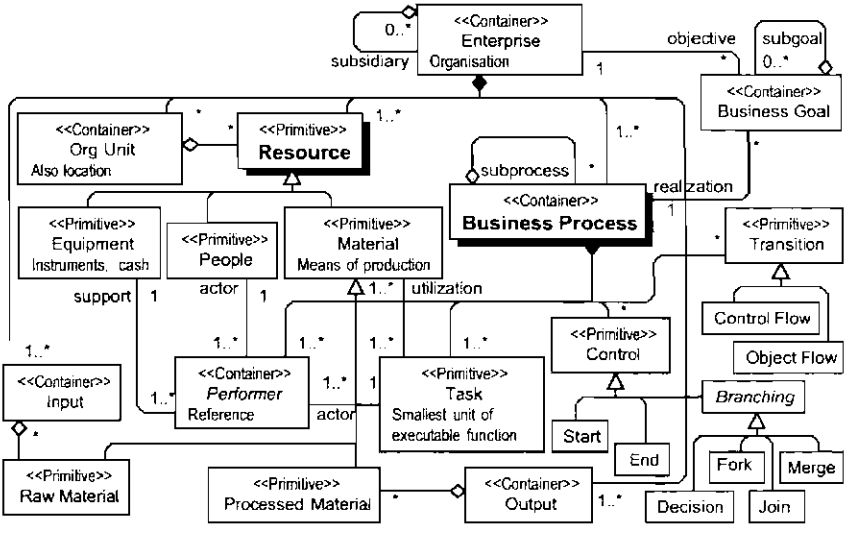


Fig. 4 Business Process Metamodel (M2)

A common view on business processes has been already studied in [11]. In the current research, the main interest is directed towards the dynamic behavior of an enterprise, therefore, other concepts are included, only if they play a role for business processes. It actually is a cut (and renamed) version of the UML activity diagram metamodel, but with the resource management aspect added.

In this metalayer all classes are "instance classes" again, because they are specific instances for the more abstract metametalayer (M3, Fig. 7).

A business process is developed as a *realization* of some *Business Goals* in *Enterprise*. Each of the Enterprise has *Resources*, *Organizational Unit* and *Business Processes*. *Material* is one of *Resources* that belongs to *Organizational Unit*. *Raw*

Material flows from *Input* of *Enterprise* to *Output* and becomes to *Processed Material* through *utilization* in a *Task*. *Task* is an atomic part of *Business Process* and doesn't contain anything else (i.e. elementary action in UML AD). A *Task* has a special reference - *Performer*, which is used for indirect pointing to a *Resource* (such as *Equipment* and *People*), that performs or is necessary for the task execution. The amount of *Material*, that is utilized in a task, is decremented after the task execution. The count of free *People* and *Equipment* is used only (is decremented) during task execution time, and becomes free (is incremented) after.

Controls are used for controlling the task execution sequence (branching - parallel processing and decisions). *Transition* (*Control Flow*) shows task execution sequence, but *Object Flow* additionally shows what object (message or physical material) is sent from task to task (as in UML AD).

6. Measure Definition Metamodel (M2)

In addition to the described business process metamodel (section 5), a harmonized measure declaration metamodel (M2 layer) is derived. This metamodel (Fig. 5) provides standard declarations for the most typical measures. This model again uses the "instance class" notation with respect to the metameta layer (Fig. 7). The main value of this metamodel is in the fact that it shows how process element structuring translates into appropriate measure value aggregation for the most typical measures.

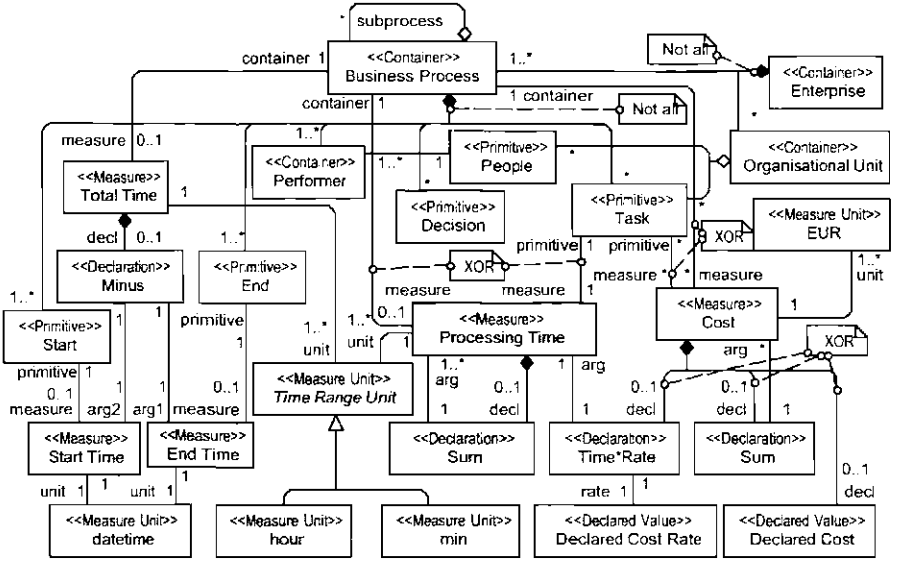


Fig. 5 Measure Declaration Metamodel (M2)

In the metamodel it is defined, how the selected measures are related to business process objects and to each other. If a measure has a declaration expression (an

implicit one or to be explicitly defined in a model), this expression is shown as a function-arguments complex, the function having the *<<Declaration>>* stereotype. In Fig. 5, it is shown, which business process elements can have *Start Time* and *End Time* measures. Some process elements e.g., *Business Process* or *Task* can have several measures, such as *Processing Time*, *Total Time* or *Cost*. A declaration for a measure depends on the process element to which it is attached and how it is declared (when explicit declaration in a model is possible). E.g., *Cost* can be declared either as an exact value (*Declared Cost*), as a multiplication of *Processing Time* by *Declared Cost Rate*, or as a sum of several sub-costs (the only possibility if the *Cost* measure is attached to a *<<Container>>* object - e.g. *Business Process*).

Processing Time can be without declaration (if the value is obtained directly – it is attached to a *Task*), or can be a sum of several *Processing Times* (if the measure is attached to a *Business Process* object). A measure can have several *Units*. In this example, several units (e.g. *hour*, *min*) are shown only for the *Processing Time* measure.

This metamodel could serve as basis for completely precise measure declaration scheme, where all implicit declarations and possible explicit declarations for each feasible measure-object pair can be specified. Namely, the pair object-measure makes the declaration features unique, measure classes in Fig. 5 are "reused" to reduce the diagram size. All this could be specified formally as standard OCL [14] constraints at measure classes.

7. Measure Classification and Relation to Business Objects (M2)

Through expanding of the partial measure declaration metamodel (Fig. 5), a complete metamodel is obtained. Possible assignments of measures to business process elements (from Fig. 4) are shown in Fig. 6.

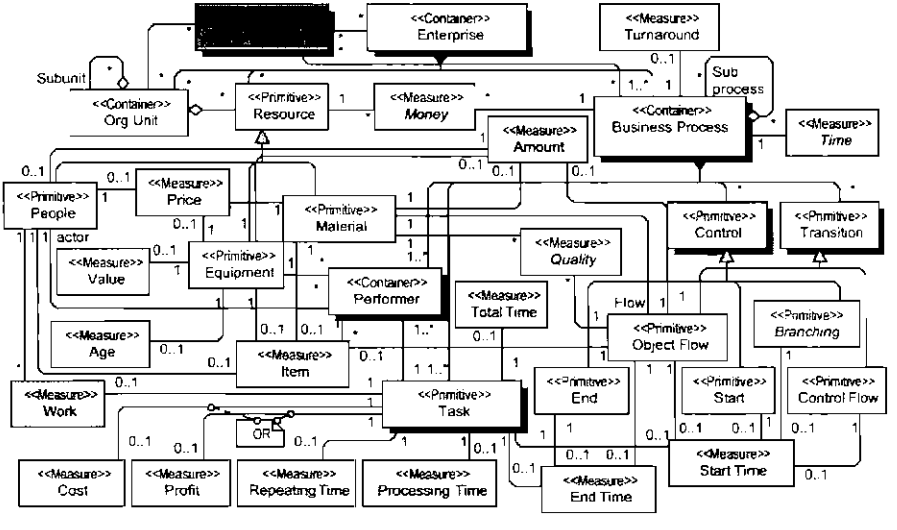


Fig. 6 Measure Relations to Business Objects Metamodel (M2)

Measures used in typical process modeling tools [7,9,10] actually are subsets of the assignment provided here. Therefore the metamodel in Fig. 6 could serve as a certain standard. However, since another useful measures can be always be invented, a general framework for assigning any measure in a similar way to any process element (business object) will be described in section 8.

Through analysis of several management approaches [2,3,4,5,6], the following measure groups were introduced: *Time*, *Money*, *Resource*, *Work* and *Quality* (marked as blue, green, orange, violet and azure colors). Unnamed associations are used in Fig. 6 to assign a measure (a class with the stereotype <<*Measure*>>) to a business object (<<*Primitive*>> or <<*Container*>>).

The provided metamodel has a significant value, since attaching any measure to any process element would be a semantic nonsense. Measure declarations, constraints and aggregation functions are out of scope of the current paper, therefore it will not be described in details here.

8. Business Measure Metametamodel (M3)

To measure and analyze business in a comprehensive way, a generic and common methodology for all possible business management areas is necessary. Therefore one more abstraction layer or metametamodel (M3) should be introduced. I.e., building a tool on the basis of a more abstract layer provides the possibility to add new measures for objects or change existing.

According to MOF traditions, the metametamodel (the MOF M3 layer) is kept simple and fixed, and all the complexity of a specific domain should be represented by its metamodel. This would imply an intensive use of OCL constraints for a metamodel of a complicated domain, in order to specify the intended semantics. Here another solution has been tried (a legal one with respect to MOF standards), the metametamodel is extended by specialized classes, in order to specify semantic constraints for a set of metamodels in a readable way.

In the proposed metametamodel (Fig. 7) the new metaclasses *Business Object* and *Measure* are defined as specializations of *Class* from UML *InfrastructureLibrary::Constructs*. The *child* association is meant to be the same one from UML metametamodel, which associates a class as a part of another (actually the real metametamodel is more complicated there). In this way we can retain the UML metametamodel for the "modeling part" of metamodels, and have a framework for modifying types of measures. The main aspect we want to know from this specific metametamodel is, which metaclasses (business objects) represent the primitive model elements and which the container (composite) ones. The metamodels (Fig. 4,5,6) can be obtained as "instantiations" of the proposed metametamodel, using the respective <<*Primitive*>> or <<*Container*>> stereotypes for business objects.

According to the metametamodel (Fig. 7), a *Measure* is a concept, whose main part is its *Declaration*. The measure *Declaration* **can** contain a *Declared Value* (a constant), or several *Math Functions*, which use *Declared Values* or other *Measures* as argument (*arg* role). If a *Business Object* is a *Container*, the corresponding

Measure **must** contain *Aggregation Function*, which uses other *Measures* that belong to the contained *Business Objects* as a source.

A measure *Declaration* can reference another measure in two ways – either explicitly a named measure (e.g., *Processing Time* in Fig. 2), or implicitly through *Aggregation Function*, if the measure is attached to a *Container* object (e.g., *Cost* for *Make Order* in Fig. 1). A *Measure* references also one of several possible *Measure Units*.

Each measure is associated to some *Business Object* (a relevant element of the business process metamodel). *Business objects* can be either *Primitive* (e.g. Task), or *Container* (e.g. *Business Process*). If a business object is a *Container*, and its measure has no *Declared Value*, then values of this measure **must** be calculated using the specified values from *child* (contained) objects.

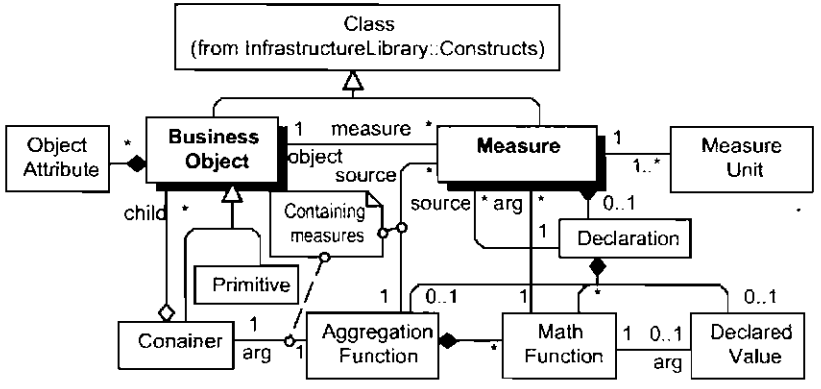


Fig. 7 Measure Metametamodel (M3)

9. Business Process Execution and Measure Values

Measures itself are only declarations or definitions of what and how some business objects will be measured. As it was described in section 4, actual values are obtained only at system runtime. For declaration execution, an exact semantics dependant only on the measure declarations, but not on the way the process is run, should be defined.

Existing process management and simulation tools each have a different definition of process semantics. On the basis of a common framework for business process modeling [11] a common framework for process execution semantics could also be defined, but details are well beyond the limits for this paper.

Assume that the some execution engine executes processes, shown in Fig. 1, 2, and provides execution log shown in Tab. 2. In the following table, in contrast to Tab. 1, **row represents a dynamic instance of an each model element** (i.e. including instances of "system" elements). Cells in columns *Object Type* and *Object* represent the object (diagram element). Cells in *Start Time*, *End Time*, *Processing Time* columns show the actual values of instance attributes. *No* column is a unique instance (row) identifier in the table.

The most important for value aggregation is the *Process ID* column (new Token ID or a new process copy identifier). The new unique ID is generated for each **new** process execution, when business process starts in its start point in the top level business process. In the proposed notation the start point is declared explicitly, but it could also be declared implicitly. This process ID is used for all process elements till the process instance end.

No	Object Type	Object	Process ID	Start Time	End Time	Processing Time	Performer
10023	Business Process	Sell Pizzas	00101	9:05:34	10:15:35		
10024	Business Process	**Make Order**	00101	**9:05:34**	9:15:47		
10025	Task	Fill Order	00101	9:05:34	9:06:03	0:01:31	
10026	Business Process	Sell Pizzas	00102	9:06:12	10:20:12		
10027	Business Process	Make Order	00102	9:06:13	9:08:02		
10028	Task	Fill Order	00102	9:06:13	9:07:01	0:01:12	
10029	Task	Check Order	00102	9:07:01	9:07:48	0:00:47	Clerk2
10030	Decision	Need Correction	00102	9:07:48			
10031	Task	Make Corrections	00102	9:08:57	9:16:05		
10032	Task	**Check Order**	00101	**9:10:03**	9:13:03	**0:03:00**	Clerk1
10033	Decision	Need Correction	00101	9:15:39			
10034	Task	**Send to Production**	00101	**9:15:40**	9:17:40	**0:02:00**	Clerk1
10035	Task	Check Order	00102	9:16:10	9:17:48	0:01:38	Clerk2
10036	Decision	Need Correction	00102	9:17:48			
10037	Task	Make Corrections	00102	9:17:59	9:20:00		
10038	Task	Check Order	00102	9:16:10	9:17:48	0:01:38	Clerk2
10039	Decision	Need Correction	00102	9:16:12			
10040	Task	Send to Production	00102	9:20:01	9:21:02	0:01:01	Clerk2

Tab. 2 System Runtime Log Example

This execution log is used for extracting measure values. Rows in Tab. 2 that are used for the generation of the measure value example in Tab. 1 are marked yellow, and the attribute values that are really used for the measure value calculation are shown in bold rectangles.

For container objects such as business processes the measure aggregation is used according to the implicit declarations specified in Fig. 3 and definition in Fig 2. The aggregation is always performed for the same process execution instance, using *Process ID*.

The described principles are sufficient for a formal definition of a universal measure value extraction procedure. Such a procedure would provide the precise measuring semantics. However, there may be more complicated situations within the described framework. E.g., if both the main and subprocess diagrams contain loops, the provided identification at the process execution instance level is insufficient. The finding of best universal value extraction procedures is a theme for future research.

10. Conclusions

Business process management systems or even simulation experiments produce large amount of plain data, which show no clear picture about the real business process. Aggregation and analysis of these data requires development of new methods and calculations, because existing tools support only a small part of interests.

In the current paper, a new look on business process measurement problem is proposed. The problem is analyzed in an unambiguous and formal way using UML. Several process measuring methodologies are merged with the metamodeling approach according to MOF, and a comprehensive business process measurement metamodel has been developed.

The proposed approach allows defining values in a natural way, and measurement of data, which are of interest to business, without deep investigation into specific technical solutions. This provides new possibilities for business process measurement, decreasing the gap between technical solutions and asset management methodologies.

As a further research, development of a more detailed metamodel and standardization of system runtime is planned. The research results will provide a framework for metamodel-based business modeling/simulation/management tools, and will extend them with comprehensive business process measurement possibilities.

12. References

1. Gartner, Inc, Executive Report Library, "Winning Asset Management Series. Introduction and Report Overview" http://www.gartnerconnects.com/executivereports/executive_reports_asset_management.pdf
2. ISO 9000:2000, Quality management systems - Fundamentals and vocabulary , ISO 9001:2000, Quality management systems – Requirements, http://www.iso.org
3. Mark C. Paulk, et.al., The Capability Maturity Model : Guidelines for Improving the Software Process, Addison Wesley Professional, 1995
4. Douglas T. Hicks. Activity-Based Costing : Making It Work for Small and Mid-Sized Companies, 2nd Edition, John Wiley & Sons, 2002
5. Robert S. Kaplan, David P. Norton, The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment, Harvard Business School Press, 2000
6. Farok J.Contractor,Valuation of Intangible Assets in Global Operations,Quorum Books, 2001
7. ARIS 6 Collaborative Suite, System White Paper, IDS Scheer, 2003, http://www.ids-scheer.com/sixcms/media.php/1186/ARIS+6-2+SWP+en+2003-07.pdf
8. System Architect, Tutorial, 2001, Popkin Software, http://www.popkin.com/products/product_overview.htm
9. QPR Process Guide White Paper, QPR ScoreCard White Paper, 2002, QPR Software Plc, http://www.qpr.com/protected/whitepapers/QPR_ScoreCard_WhitePaper.pdf
10. Casewise Corporate Modeler Product Info, Casewise, http://www.casewise.com/products/corporate-modeler/corporate-modeler.php
11. Vitolins Valdis, Audris Kalnins. Modeling Business. Modelling and Simulation of Business Systems, Kaunas University of Technology Press, Vilnius, May 13-14, 2003, pp. 215.-220.
12. Unified Modeling Language: Superstructure, version 2.0, Object Management Group (OMG), 2003, http://www.omg.org/docs/ad/03-04-01.pdf
13. Meta Object Facility (MOF) 2.0 Core Proposal, Object Management Group (OMG), 2003, http://www.omg.org/docs/ad/03-04-07.pdf
14. Response to the UML 2.0 OCL RfP Revised Submission,Version 1.6. Object Management Group (OMG), 2003, http://www.omg.org/docs/ad/03-01-07.pdf

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Datbases and Information Systems
198 207 P.

Global Software Development Process Management: Problem Statement

Darja Šmite, Juris Borzovs

Riga Information Technology Institute,
45b Kuldigas street, Riga, Latvia
{Darja.Smite, Juris.Borzovs}@riti.lv

Abstract. This problem statement paper represents a starting point for a research in global software development. In particular: outsourcing relationship definition and management, as well as ways of software development process improvement. The lack of research in this area precludes the full understanding of how to manage outsourcing projects and parties relationship. The authors give an overview of the existing research on this topic and outline the main problems and new trends by representing a framework for future research in this area. As a result of this research the authors plan to summarize it in a set of process diagrams, recommendations and supporting tools, which will enhance the knowledge in this area and help to standardize and improve the process of global software development.

Keywords. Development outsourcing, global software development, offshore outsourcing, outsourcing relationship

1. Introduction

The question explored in this paper relates to the issue of global software development relations and its success factors. In particular: "Is there any specific methodology to manage and monitor the outsourcing relations?"

A review of research in outsourcing relations highlighted the gap of knowledge in relationship quality processes, which could manage and monitor the global software development processes between two IT related organizations.

To start with, the term of outsourcing has to be clarified. The concept of outsourcing can be viewed from many different views – definition of the process, process types, and related problems (contractual questions, management, risk assessment, success factors, etc.). There are various types of outsourcing:

- Business process outsourcing (BPO),
- Information system (IS) maintenance outsourcing,
- Application or application service provider (ASP) outsourcing,
- Hardware outsourcing,
- Data centre outsourcing,
- Selective or full software development outsourcing.

Some of the types are widely known and there are world wide consulting firms which provide their clients with information on the different views of the topic, e.g.

contractual questions, decision making, risk assessment, vendor ratings and statistics.

The questions of software development outsourcing processes, relationship management and quality assurance are poorly discussed in the related literature, and remain a relatively unexplored topic that demands examination in depth.

The majority of research is devoted to exploring business process or IT related process outsourcing in general or for non-software development companies. Most of researches bring up the questions of decision making – whether to outsource ([12], [31], [34]), relationship risk management ([3], [5], [6], [10], [30]), contractual problems and advices ([4], [7], [12]), success factors that will help to survive starting outsourcing relationship ([16], [20], [21], [26], [30]) and case studies from the field ([9], [21], [25], [27]). Outsourcing as a trend in software development process improvement is still being not clearly understood and require research in-depth. One of the key pointers to achieve a productive partner relationship by Jae-Nam Lee is to "define, agree, and communicate clear and measurable standards of performance" [24].

What is software development outsourcing? How do the involved parties collaborate? How is the development process managed? How can the remote process be improved? What are the core success factors related to remote software development outsourcing? How to improve and regulate quality assurance in outsourcing projects? This appears to be a simple set of related questions, to provide answers to.

Only some of researchers bring up the answers to the questions of distance and communication improvement between outsourcing partners ([10], [21]), organization and allocation ([13], [28]). There is only a handful of research in IT outsourcing highlighting the fact that the elements like shared knowledge, mutual dependency, organizational linkages and cultural unity are also important and influence the success of the relationship [13], [23], [25]. According to Christof Ebert and Philip De Neve it seems to be obvious that all development locations working in one product line use the same processes, methodology, and terminology even when changes occur, but in an organization with several thousand engineers, separated thousands of kilometers from each other, having different languages and cultures, it is quite a challenge [13].

Regardless of the gap in related literature, the topic of software development outsourcing is the top-line question. Two statements from recent Gartner research support this hypothesis:

"The use of external services providers e-enablement and other ... competencies will double by 2005, despite the increasing complexities of the supplier market (0.7 probability)".

"By 2003, a majority of Global 2000 enterprises will use externally sourced workers to handle more than half of their application delivery work (0.8 probability)" [26]

The enterprises being the customers are seeking ways of their IT development process cost reduction. They are often answered "Transfer your development offshore". Yet, how can the outsourcing relationship be managed effectively still reducing the cost of management is the question that needs the answer. The aim of this research is to uncover the process of resource allocation and management in global software development alliances, communication improvement, system

development process management, delivery quality assurance methods and final customer satisfaction.

The authors' purpose is to begin research by showing the topic's rich potential and to highlight unrecognized questions and problems.

The broad objectives of this research can be characterized as follows. The first objective is to mark and classify all the possible software development outsourcing models and to evaluate the models' perspectives. The second objective is to work out the guidelines, recommendations and methods for outsourcing software development relationship management in order to facilitate successful software development and end-consumer satisfaction.

The paper is organized as follows. The following section reviews the literature related to software development outsourcing definition. Next, the authors discuss the trends and motivation of their research in this area. The research approach and framework for further research are offered then. The paper ends with a brief summary.

2. Outsourcing overview

2.1. What is outsourcing?

Outsourcing is not a phenomena or a new trend. It is known since early 60s-70s focusing on hardware and software outsourcing [24]. Outsourcing is typically defined as a practice of transferring IT related processes from internal IT functions to third-party vendors. The scope of outsourcing varies from IT assets, leases, and staff to management responsibility for delivery of services [18].

The authors' interests are tied to selective or full software development outsourcing from other countries, which is known also as Global Software Outsourcing (GSO). According to R. Heeks global software outsourcing is the outsourcing of software development to subcontractors outside the client organization's home country [17].

The main motivation to outsource is produce products more effectively, reduce time to market and project costs. According to K. Bodker outsourcing has been one of the most influential factors contributing to changes in information systems development in the last decade [7]. Clearly, the lack of research in this area precludes the full understanding of how to manage these relations and how to succeed in them.

2.2. Partnership management

Outsourcing in the global space with rapid changes is not a trivial process. The distance between the outsourcing parties and their probable culture differences are the major obstacles that complicate communication between the parties. Outsourcing vendors often work with more than one customer. That's why it makes difficult for the customer to manage the vendor's organization and the related processes, which are managed by the vendor's internal rules. The evolution of outsourcing relationship now has a tendency of partner-based outsourcing. So called strategic

alliances become popular because of limitations of legal contracts and convenience of partnership establishment [11], [15], [25], [26], [29].

The acceptance of strategic alliances among business partners is good evidence for the growing popularity of partner-based outsourcing [22].

Strategic alliances give the customers opportunities to establish the directions of development as well as manage and subordinate the partner outside. It allows implementing a better relationship management, concerning the strategies to accomplish desired performance goal and improve the communication and information share. Alliances allow a firm to leverage a key part of the value chain by bringing on a strong partner that complements its skills, and to farm out processes the company is not good at, and to create an opportunity to innovate [25]. This sort of relationship brings such important factors as mutual dependency, trust and security.

Figure 4. Global software development. Outsourcing to IT factories worldwide.

This tendency rises up a new concept of the out side development outsourcing – software development factories, which are governed by the alliance head office.

Considering this trend the whole process of software development can appear in a number of new different models, which are not understandable and standardized yet.

The software development process can be shown as a puzzle with many different separate tasks (Figure 2).

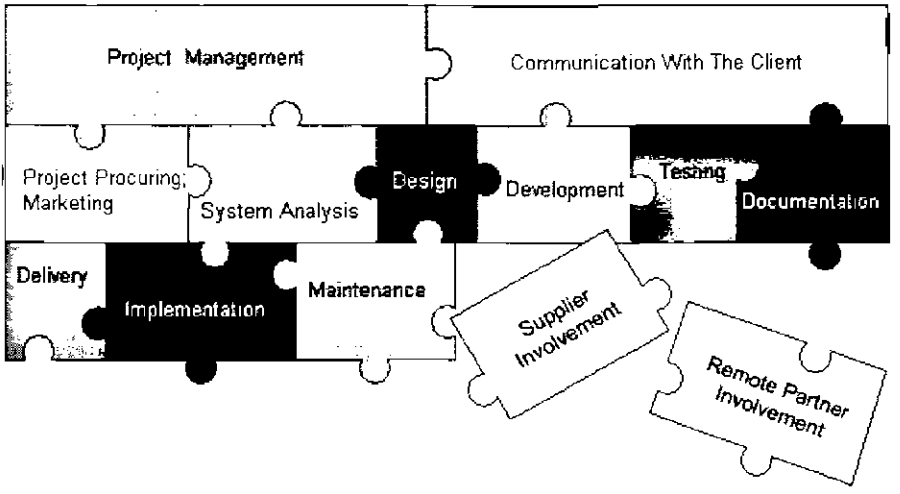


Figure 4. Software development tasks.

The distance between remote partners in a global alliance may cause some unexpected changes in these yet well known and standardized processes. The most difficult question is the Supplier and Remote Partner involvement in the project management and communication with the client as well as in the rest of the tasks. The question of resource allocation, the most effective global project management model, the question of how and where to maintenance the products – all these questions need the answer.

2.3. Technology

Recently, a new term in the field of outsourcing management has been discovered. Outsourcing Relationship Management or ORM is the term used to describe the software tools and services necessary to successfully manage complex sourcing relationships as well as improve the sourcing value proposition for customers and vendors [33]. Benny Lehmann [33] in the interview as the most important elements of the ORM solution lists the following: compliance monitoring and analysis of service delivery objectives from outsourcing providers; financial analysis enabling tracking of budgets to actual spend, charge back allocations and measurement of incentive-based contracts; management of multiple outsourcing/insourcing relationships and the management of resources (people, hardware, software) in a widely distributed environment; and standardized management of contract issues such as breach, dispute resolution, escalation, renegotiation, etc. This innovation ignores such elements as information and communication management. Communication as well as knowledge and information share is a difficult issue due to the probable distances between the partners being nevertheless a key point in the relationship management.

Supporting software is not the least thing in the management of the outsourcing software development relationship. An appropriate, secure, effective and convenient

software tool can optimize the complex outsourcing software development management processes and enhance vendor management capabilities that ultimately reduce the cost and improve efficiency.

3. A framework for it development outsourcing relationship

In this section, the paper discusses the research background and approach for studying relationship management and process aspects of the global software development.

3.1. Case background

To reduce the gap in outsourcing process understanding, the research has to respond on both industry and academy experience. It can be achieved, if researchers have an opportunity to study real life processes in industrial environment.

As a background for the further research the authors will use several Latvian software development companies' experience in outsourcing and joint venture establishing.

IT development is a very fast developing sector in Latvia. It can be characterized by the following figures [16], [19]:

- Production of ICT goods and services was 4.6% of GDP;
- 10 000 ICT professionals;
- 251 software consultancy companies, 173 software & computing services companies;
- FDI in the sector USD 14 million;
- Software development was 0.5% of GDP;
- ~100 software development companies;
- Annual growth in software development 15%;
- Export growth 15.7%.

Software development and services in Latvia is a robust part of the industry, led by companies such as DATI, Exigen, TietoEnator, Fortech, Alise, Verdi and Tilde. Software design is the most important IT sector in Latvia. Outsourcing had become the core competence of software developers and contributes to a major part of Latvian exports. Software maintenance, integration, consulting and training are some of the most rapidly growing areas of the Latvian IT services industry [19].

It is notable that several companies have become part of large international IT networks through mergers and acquisitions. Tieto Corporation of Finland acquired Tieto Konts and has established a new subsidiary, TietoEnator, in Riga — TietoEnator Financial Solutions. SWH Technologies merged with IS company Exigen and is now called Exigen Latvia. Fortech and VAR have become part of the Estonia-based MicroLink Group. DATI Group is a long term partner for software development clients in UK, Germany, Sweden and other countries.

The IT product export in Latvia has a tendency to grow annually. The growth of IT market is remarkable and has its reasons. As advantages of IT market in Latvia the following points can be named:

- highly skilled and cost-effective professionals,
- advanced traditions in systems and software engineering,
- western business culture and geographic location,
- developed infrastructure,
- government support,
- developed educational system.

Around 20 Latvian IT companies are involved in the Latvian Information Systems cluster [14]. This represents an association which has been established to boost the competitiveness of Latvian IT service companies, and to increase their exports to international markets. As a gateway to the cluster there has been established a new virtual network of partners and customers under the name of EVITAF – European Virtual IT Application Factory [32]. This is a recent project which will provide distributed software development and maintenance services using the global market.

It should be emphasized, that EVITAF develops custom-built systems and solutions, and does not distribute standard products. That's why the EVITAF project currently needs a framework for successful global software development management. This appears to be an industrial claim for the research. The authors' activities based on the topic of global software development will provide the theoretical background for EVITAF project that would be examined in practice.

3.2. Research approach and further activities

When dealing with information systems, human and social factors are always present. In such conditions, quantitative analysis by means of logic and mathematics are no longer appropriate. The research approach, chosen for this global development examination is case study.

The first step as a starting point for the research would be to gather information about the existing relationship processes and management by interviewing the project managers in the several Latvian top software development companies which are involved in global software development. The intention is to inductively derive, via interviewing experienced engineers, management and end users about the different sides of relationship and product delivery, an understanding of a probable future global software development model. This will help to sketch the construct and to build a framework for the further research.

The second step in outsourcing relationship improvement would be the analysis of aspects that influence project management. Outsourcing processes can be viewed from many different facets:

- Resource based view – resource share-out, arrangement and management,
- Contractual dependency,
- Goal alignment,
- Relationship management with external customers,
- Software development process share-out, organization and management,
- Communication improvement,

- Information and knowledge resource allocation and share,
- Project official registration – process documents for a better evidence and quality control,
- Quality auditing.

This might uncover the necessity of supporting tools for the more effective relationship management. The information will be gathered by interviewing and mediatizing. Considering the listed aspects will help to build a foundation for a stable relationship management framework.

All the processes have to be described then in form of process diagrams, guidelines, recommendations and methods of managing outsourcing software development relations. Supporting tool development may be processed by this time as well.

As a checkpoint for this model it has to be tested on real projects. This matches to be the next step of the research. The results of the testing will be used for the framework elaboration and improvement.

Finally after these steps global software development processes have to be understood, described and summarized in a clear model provided with guidelines, recommendations and methodologies.

4. Conclusions

The decision to start such a research project was advocated by the problem topically. Today there is a big gap between theory and practice or between academy and industry in terms of global software development. The volume of the globally correlating society is huge and continues to grow. The lack of literature and research in this area precludes the full understanding and improvement of the global software development.

This problem statement paper deals with sketching a framework for the further global outsourcing development research and its motivation. Based on a case study and the academy look the framework has the potential to further our understanding of one of the most popular research topic in software development improvement. By describing all the necessary procedures, methods, and providing recommendations it will bring the model to a complete condition as well as will improve the foundation for outsourcing software development.

5. Acknowledgement

This research on Global Software Development Process Management is partly supported by the Latvian Council of Science project Nr. 02.2002 "Latvian Informatics Production Unit Support Program in the Area of Engineering, Computer Networks and Signal Processing".

References

[1] Aman, A. and Nicholson, B. The Process of Offshore Software Development: Preliminary Studies of UK Companies in Malaysia. Information Systems Perspectives and Challenges in the Context of Globalization 2003, pp.201-216

[2] Arnett, K.P. and Jones, M.C. Firms that Choose Outsourcing: A Profile. Information & Management, 2` (4), 1994, pp.179-188

[3] Aubert, B.A.; Dussault, S.; Patry, H.; Rivard, S. Managing the Risk of IT Outsourcing. CIRANO Working Papers, Montreal, June 1998

[4] Aubert, B.A.; Houde, J.F.; Patry, H. and Rivard, S. Characteristics of IT Outsourcing Contracts. Proceedings of the 36th Hawaii International Conference on System Sciences, HICSS'03

[5] Aubert, B.A.; Patry, H.; Rivard, S. and Smith, H. IT Outsourcing Risk Management at British Petroleum. Proceedings of the 34th Hawaii International Conference on System Sciences, 2001

[6] Bahli, B.; Rivard, S. A Validation of Measures Associated with the Risk Factors in Information Technology Outsourcing. Proceedings of the 36th Hawaii International Conference on System Sciences, HICSS'03

[7] Bodker, K., Is Development in an Outsourcing Context – Revisiting the IS Outsourcing Bandwagon.

[8] Buchowicz, B.S. A process model of make vs. buy decision-making: The case of manufacturing software. IEEE Transactions on Engineering Management 38, 1(1991), pp. 24-32

[9] Carmel, E.; Agarwal, R. Tactical Approaches for Alleviating Distance in Global Software Development. IEEE Software, March/April 2001

[10] Clemons, E.K.; Hitt, L.M. and Snir, E.M. A Risk Analysis Framework for IT Outsourcing. Draft, 2000

[11] Currie, W.L.; Desai, B.; Khan, N.; Wang, X.; Weerakkody, V. Vendor Strategies for Business Process and Applications Outsourcing: Recent Findings from Field Research. Proceedings of the 36th Hawaii International Conference on System Sciences, HICSS'03

[12] Department of Information Resources, Austin, Texas. Outsourcing Strategies: Guidelines for Evaluating Internal and External Resources for Major Information Technology Projects. June, 1998

[13] Ebert, C.; De Neve, P. Surviving Global Software Development. IEEE Software, March/April 2001, pp.62-60

[14] Feists, V. A Review: IT Cluster in 2001 and Export Potential in Latvia. Baltic IT&T Review, No.2(25), 2002, pp.52-55

[15] Goth, G. The Ins and Outs of IT Outsourcing. IT Professional, January / February 1999

[16] Grover, V.; Cheon, M.J.; and Teng, J.T.C. The Effect of Service Quality and Partnership on the Outsourcing of Information Systems Functions. Journal of Management Information System, 12 (4), 1996, pp.89-116

[17] Heeks, R.; Krishna, S.; Nicholson, B.; Sahay, S. Synching or Sinking: Global Software Outsourcing Relationships. IEEE Software, March/April 2001, pp.54-60

[18] Hirschheim, R.; Lacity, M. The Myths and Realities of Information Technology Insourcing. Communications of the ACM, February 2000/Vol.43, No.2, pp. 99-107.

[19] International Trade Centre UNCTAD/WTO. Global Technology Markets: Country Export Potential Profile – Information Technology: Latvia 2002. Geneva: ITC, 2003.

[20] Lacity, M. Lessons in Global Information Technology Sourcing. IEEE Computer, August 2002, pp.26-33

[21] Lacity, M.C.; Willcocks, L.O.; and Feeny, D.F. IT Outsourcing: Maximize Flexibility and Control. Harvard Business Review, May-June 1995, pp.84-93

[22] Lee, J.N. and Kim, Y.G. Effect of Partnership Quality on IS Outsourcing Success: Conceptual Framework and Empirical Validation. Journal of Management Information Systems 15, 4 (1999), 29-61.

[23] Lee, J.N. and Kim, Y.G. Exploring a Causal Model for the Understanding of Outsourcing Partnership. Proceeding of the 36th Hawaii International Conference on System Sciences, HICSS'03

[24] Lee, J.N.; Huynh, M.Q.; Kwok, C.W. and Pi S.M. IT outsourcing evolution: past, present and future. Communications of the ACM, May 2003/Vol.46, No.5, pp.84-89.

[25] Lee, J.N.; Huynh, M.Q.; Kwok, C.W. and Pi S.M. The Evolution of Outsourcing Research: What is the Next Issue?. Proceeding of the 33rd Hawaii International Conference on Systems Sciences, Maui in Hawaii, January 2000, pp.1-10.

[26] Light, M. Matlus, R. Berg, T. Strategic Analysis Report Application Development Contracting: Lifeline or Noose? R-14-38791, 28 September 2001

[27] Loh, L. and Venkatraman, N. An Empirical Study of Information Technology Outsourcing: Benefits, Risks, and Performance Implications. Proceeding of the 16th International Conference on Information Systems, Amsterdam, the Netherlands, December 10-13, 1995, pp. 277-288.

[28] Meadows, C.J. Globalizing Software Development. Journal of Global Information Management, Vol 4, No.1, 1995

[29] New Strategies in IT Outsourcing. Case Study: CRESTCo Ltd.. Business Intelligence 1999

[30] Report of Latvian Ministry of Economics, June 2002

[31] Roy, V.; Aubert, B.A. A Resource Based View of the Information Systems Sourcing Mode. CIRANO Working Papers, Montreal, October 1999

[32] Sukovskis, U. Quality – Key Factor in the Valtic Market for IT Outsourcing. Presentation at Baltic IT&T 2003 Forum: eBaltics, April 2003.

[33] The Outsourcing Institute. Q&A with Digital Fuel CEO, Benny Lehmann by Frank Casale, Chairman & CEO of the Outsourcing Institute.

[34] Willcocks, L.; Fitzgerald, G. Guide to Outsourcing Information Technology. Business Intelligence, 1994

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Datbases and Information Systems
208 219 P.

Experience of Introducing a Metamodel-based Traceability Tool into Software Development Projects

Martins Gills

Riga Information Technology Institute (RITI)
Kuldigas iela 45, Riga, LV-1083, Latvia
martins.gills@riti.lv

Abstract. This paper describes an experience of introducing a dedicated traceability tool TraceIt into projects at IT company. The tool supports impact analysis, testing and problem reporting processes. It was developed as part of process improvement initiative at IT company and was designed to be configurable for each project by means of the traceability model (metamodel). The usage of TraceIt was grown from a pilot study to a tool that is systematically used in projects of IT company. In this paper, the typical process of introducing the tool into software development projects is shown. The decisions of project staff and associated problems are analyzed. Special issues related to metamodel specifics are reviewed. Current results encourage a further development of the tool.

Keywords. Traceability tool, problem reporting, testing process

1. Introduction

Software development projects systematically experience changes in requirements, design or implementation. Also, testing introduces different test sets and generates a list of defects. All that forms base for unlimited set of software development artifacts. It is well known that practically every software artifact could be subject to a formalized traceability [7].

The aim of the paper is to present the experience of introducing the traceability tool that was developed as part of process improvement initiative in a software development company. One of early and useful lessons learned was the idea to use the tool for wider variety of tasks than just simple traceability [6].

For this research, there was no intention to obtain formal metrics of how the tool influenced the productivity of projects. Rather, the focus was on individual experience of project team members and theirs evaluation. Each new project had both different requirements for traceability process and they experienced different stages of TraceIt development (capabilities of the tool improved over the time). Currently, the experience of tool usage accounts 3 years and at least 14 projects where 12 of them are reviewed in this paper.

This paper has the following structure: Section 2 describes the context of developing a new traceability tool. In Section 3 the proposed improvements are described. Section 4 reflects experience of applying the tool in projects. Lessons and implications are given in Section 5. The comparison with related research works is

given in Section 6. Finally, in Section 7 author draws the conclusions and outlines future works.

2. Observed problems with traceability

The motivation of developing a traceability tool came from the industry. One of the largest software developing companies in Eastern Europe sought solutions for improving its software development processes. The profile of this IT company is development of custom information systems. One of interesting features at IT company was that traditionally there were projects of different staff size and duration. Typical large projects involved about 100 developers and lasted for at least one year. The smallest projects were formed from just two developers and ran for less than half year. Another distinctive feature was that each project applied somehow different methodology and project life cycle. Also, they had different development environments. As the company followed the TickIT principles [4] and the international standards like J-STD-016 [9] there was a strong aim to maintain the traceability at high precision level.

Requirements in observed projects were typically in word processing documents. Some basic traceability between various parts of documents was maintained in a spreadsheet format. Well-established traceability matrices required a skilful person to maintain all the relations, and to perform all the analysis. Therefore, from the practical point of view the traceability as property of the project remained weakly developed and ineffective. This was one of the focus areas for possible process improvements within the IT company.

As there was no good solution found of how to improve the usage of the spreadsheets, one of ideas at Riga Information Technology Institute (RITI) for IT company was to develop a simple database that could hold the very basic traceability information for any given software project. The IT company had a strong intention that any proposed change in project processes had to give some positive effect on project performance and quality. As the idea of the traceability database was very vague, there were a lot of open questions of whether it is worth developing special software when there are different requirements management tools available, and many of them support the traceability. For example, DOORS [17] is has strong capabilities of requirements management, requirement version control and requirements traceability. At the same time, not projects saw this tool as useful means for semi-formal project process. Also, the RequisitePro [8] was seen as tool requiring the adherence to particular pre-defined process. Many tools promise the traceability support, but in many cases it is more an added feature to some other functionality, and this addition cannot be efficiently used. Most of projects did not use special requirements management tools, and there were many individual reasons of why any particular tool could not be introduced in the given project.

3. Proposed improvement via traceability tool

3.1. Pilot project

A pilot project TraceIt was started to explore the possibility to create a simple and usable traceability database. One of the first questions asked by projects was about the reasons to start using some new tool. Also the possibilities of integrating TraceIt with other tools or replacing them were analyzed. It became clear quite early that such a tool could not replace text documents or modeling environments. Rather, its task could be to keep IDs and titles/names of requirements, functions, tests, and files. In IT company, every project used its own problem reporting environment - ranging from simple spreadsheet solutions up to commercial products like PVCS Tracker [12] and in-house developed databases. Therefore, one of further ideas was to design appropriate capabilities within the tool itself.

3.2. Metamodel

As one of requirements for the new traceability tool was that is to be adaptable to project conditions, the behavior of the tool was made dependant from traceability model there. Traceability model is a metamodel that could be represented into form of a diagram. Its two main constituents are item types and relations between them.

In principle, there could be an approach of whether to identify forwards, backwards traceability or both of them [7]. In the graphical representation of metamodel, author proposes to illustrate backward trace links because it corresponds to reference/pointing direction. At the same time, by means of the tool it is possible to follow the trace in both directions. Figure 1 illustrates two examples of traceability model for two different projects. In case (a) Requirements are derived from Customer proposals and Change requests, Tests and Software objects depend on Requirements, Problem reports have reference to the Software object and the Test. In (b), The Requirements in similar way is dependants from Initial Requirement and Change request, the Design is based on Requirements, and the DB Entities and Functions are based on Requirements, plus Methods are part of Functions in the code.

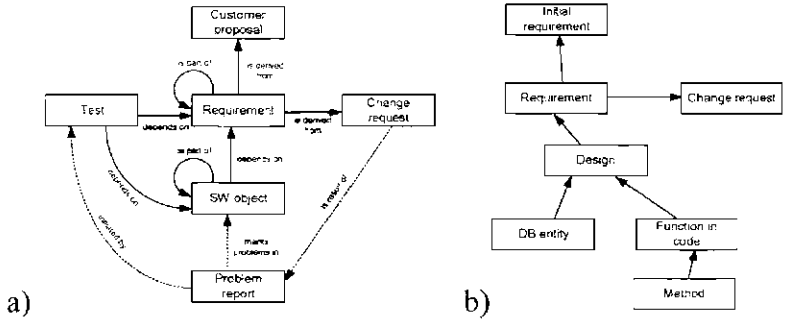


Figure 1. Two sample traceability models.

Item type is any software development artifact. Depending on each individual case it could be textual requirement, use case, diagram, document, test, test result, problem report, development note, etc.

Along with the traceability model for the metamodel of the tool there was set an imperative principle that every artifact has to have an ability to be related to some other artifact. At the same time, it is allowed that there exist items without any relations. Each metamodel is generally intended to be used only for one project. Traceability models can be re-used when two projects have similar processes.

3.3. Overview of TraceIt

Previously mentioned traceability tool TraceIt is implemented as web-based application. The code is written in PHP language, and it uses MySQL database. Because of the chosen free-ware approach the software can be used in a number of operating systems.

To be more usable in different project use cases, tool is designed to represent the same information in various forms. For example, project items can be viewed either as part of list (Fig. 2), as an individual record (Fig. 3), or as an element of tree (Fig. 4).

4. Practical use of TraceIt

The experience is described in the context of three aspects - project size, project scope (tasks) and involvement of individual project member. First, one has to point out that not all projects within IT company started using TraceIt after it was proposed. Also, at the very beginning the author was not sure enough about the capabilities of the tool to provide it to large-scale projects.

Further section reflects a study from 12 projects where the TraceIt was introduced. They differ in size, scope. Also, the staff differently used the tool depending on their role in project. In general, the initial target groups of users were small and medium size projects (up to 15 people).

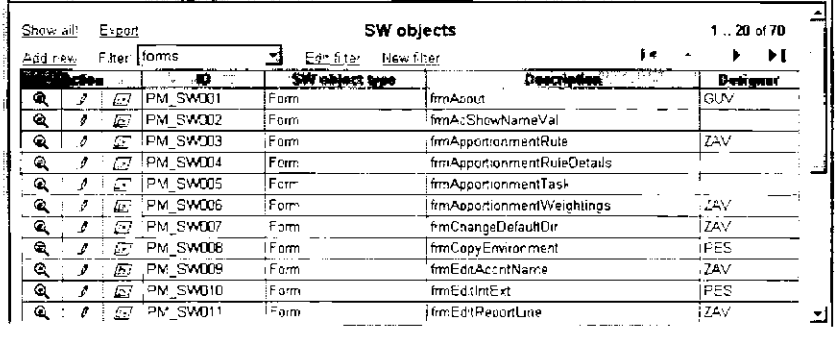


Figure 2. TraceIt screen. List of items.

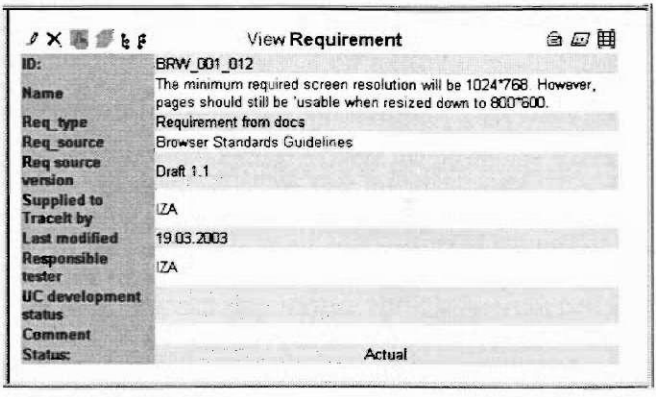


Figure 3. TraceIt screen. Individual item.

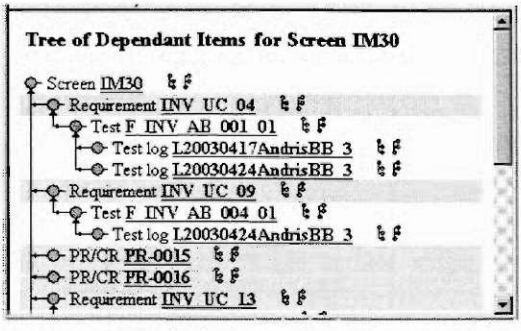


Figure 4. TraceIt screen. Relation tree.

4.1. Common process of introduction

In IT company, most of the projects were developing new software and considerably smaller proportion of the projects dealt with maintenance or performing the independent testing activities. Main type of software developed in the company - client-server architecture solutions. The typical life cycle of projects models can be characterized as incremental or cyclical.

The introduction process for TraceIt consisted of a number of common activities:

1. General information exchange between tool expert and project staff about the processes in the project takes place.
2. Tool expert prepares the initial traceability model.
3. Project reviews the model.
4. Adjusted model is configured into the tool.
5. Tool functions according to model.
6. Upon necessity, the model is updated.

Introduction was on voluntary basis, and as one of escape ways was indicated the possibility to export all information from database into spreadsheet format.

One could outline as a positive result the fact that none of projects abandoned the usage of the tool after starting using it. There are projects that have completed their tasks, and thus the usage of TraceIt was finished, too. There were some projects

which did not start using TraceIt due to project management decisions or major organizational changes. In every case where the tool was used no administrative force methods were used. Every project made its own decision. Many of them also envisaged some contingency activities in a possible event of a major failure of the tool.

4.2. Project size

Depending on the number of people involved, one could mark some three groups of projects where TraceIt was introduced: 5 small (2-4 people), 5 medium (8-15 people) and 2 sub-large (25-35 people). Small projects ran for up to 6 months, medium had duration starting from 3 months till at least 1 year, and the both observed sub-large projects currently have been running respectively for 1 and 2 years. Also, maintenance projects are still running.

When the tool was proposed all these projects were just at the kick-off phase, and they were quite open for various methodologies. In almost all cases author acted as the only tool expert and personally participated in the introduction process. Briefly after the initial and adjusted configuration of the traceability model was set, the further activities with configuration of the tool were handled to appropriate project member(s).

It was interesting that small projects quite quickly discovered this tool as means for maintaining data with variable detail level in easy manner - they did not try to make a complicated traceability model and detailed description of items, but rather had found a place where to keep notes, events, references to documents, proposals in an ordered way, especially, when several people are working on the same issue. Alternative approaches could be to use some kind of online forums, but in this case there will be limited capabilities of traceability. Problem reporting was not seen as main feature, as for small teams considerably large portion of the problems is often passed from testers to developers in a less formal way. In two cases the projects switched from informal to detailed documenting - to follow the traceability path from contract through requirements and design till the individual functions of the classes. Such an approach requires systematic documenting and also some extra work in the case of changes.

Medium and sub-large projects were both similar in a way that they were not interested to maintain all the traceability till the code function level by means of this tool. Main focus was on managing requirements and design items. Special attention was paid to the information about individual responsibilities, information sources and event timing. Problem reporting was seen as very important.

As far as it was noted, the project size does not mark radical differences, except following:

- Smaller projects were more responsive to try something new, more open to risks;
- Larger projects focused more on issue of handling a large amount of project data, and they do not like to risk with new methodologies. They tended to register and document more formally extensive amounts of information.

4.3. Project scope

Projects where TraceIt was used could be divided into three distinct categories: complete development (7 projects), maintenance (3 projects) and independent testing (2 projects).

Complete development presumes a life-cycle that spans from requirements analysis down till delivery of functional code and documentation. Typical item types that were included into traceability model within TraceIt were: requirements of various levels, design items or software objects (functions, screens and DB objects), tests and problem reports. Change requests typically were treated as a special case of problem reports. In some projects, the design items like screens were identified separately to make easier the impact analysis from the user perspective and to improve the quality of problem tracking.

Projects reviewed also the possibility to include inside the model some references to code at different levels (file, class method). The first experiments of doing this extra work proved to be little effective as during the development the code changes very often. More practical was the version control and configuration management of the code by means of dedicated tools. But at the same time, they generally provide only limited traceability capabilities. The traceability between the code and the defined software object was generally by name, and in many cases that was sufficient.

Maintenance projects usually have stable requirements, and they are more concerned with impact analysis because any requirement change may provoke a number of changes in code, tests and documentation. The observed maintenance projects included the reference to code into their traceability models. Changes occur at regular intervals, and it was found reasonable to document the relationships in TraceIt database. Important are links with change requests - they act as source for the whole set of changes. Motivating issue for detailed model was the fact that maintenance tasks time to time are handled from one team to another. In this case, important is to have a detailed project documentation.

Independent testing usually regards the software as a black box. Also, in these two observed projects there were only functional tests performed. This led to following item types: requirements/functions, screens, configurations, tests, test results and problem reports. Testers benefited from the possibility to keep all the information inside the tool. Both observed testing projects did perform mainly manual tests. If there had been some automation tools it could have raised some problems with script location and maintenance. In this case, it could be better to use the repository of testing tool.

Because no problems were observed, one can conclude that the introduced traceability tool was suitable any of the above mentioned project types. There could be some analysis made whether it is appropriate to use TraceIt as the only tool, e.g. for requirements analysis or reverse engineering.

4.4. Project member involvement

Software projects typically involve a number of project roles. Each of the project members has its own interest on traceability. We could outline the following project roles in a case of TraceIt usage:

- Project manager,
- System analyst / designer,
- Programmer,
- Tester.

Project managers typically do not analyze each individual item relation, but rather are interested to see the overall picture, the trends. Whereas the project manager for a small project usually performs also some other project role, larger projects do have a dedicated project manager. In TraceIt context, these persons had the following the areas of interest: analysis of filtered lists of items with summary data, review of statistics, item relation to personnel, follow-up of weekly or daily progress.

System analysts and/or designers are primarily involved in the traceability process. They analyze, define and transform requirements and design items. From one point of view, in small or medium size projects one or two persons know all the system properties, and they would be little interested to formalize it also into tool. The practice showed that the quality of data mainly depends on individual willingness of the person.

Within current experience, programmers had limited specific interest. They used tool for general analysis. The reduced special interest was mainly due to previously mentioned code version control separation from TraceIt. The most extensive users were testers - project size and scope did not play role. They were very interested in obtaining a good test coverage and in getting all the test execution data documented.

The identification alone of appropriate traceability model proved to be useful almost for any role. Project management that was not too concerned with the software artifact relationships received an additional view on the project processes.

Testers were more interested in traceability tool because it supported change control process, the problem processing and accounting of executed tests. This is very dynamical information in project process, and without TraceIt there have been observed technical difficulties of handling all this information.

5. Lessons and implications

In general, the main reasons why tool was not proposed or accepted in certain projects were following:

- Existing and running projects did not want to change their methods, tools and process;
- Both author of this paper and the programmers who participated in the implementation of TraceIt could give little guarantees about the stability and performance of the tool;
- It was considered that large-scale projects better rely on proven technologies (TraceIt was an experimental tool);
- No explicit guarantees that the tool will be supported in the future;
- Unclear issue about compatibility with other tools. Only data file exchange mechanism available;
- Underestimation of the traceability importance from the project management side.

As we can see, most of considerations are management level decisions, and do limit the experiment.

5.1. Observed assistance

Characterizing the main difference between projects where TraceIt has been used and the ones without any specific tools supporting the traceability one could identify a number of differences. As positive effects from tool usage one could name:

- easier control over project item dependencies,
- detailed impact analysis and effort evaluation,
- understanding of the project status,
- all information related to testing process can be kept within one tool,
- web-based implementation was useful for teams with distributed geography.

Users extensively used filtered lists of items, relation information analysis and the statistics. As the application performance was acceptable, there were no problems for users to perform quick search and analysis of particular item or item group. Very useful was the developed ability to configure items in a way that the whole information is located in the database. Primarily this refers to testing process. The description of tests, execution results and the problem reports were not kept outside the tool. When some projects needed to produce some formal documents the necessary data were exported.

5.2. Main obstacles in usage

Along with the positive results there were some obstacles in usage observed. As it was noted earlier, TraceIt could not solve all the project problems. Especially this was true in the very first projects where it was introduced. There was both little functionality available in the application and the author was not ready to propose the best solutions for using this tool.

In general, 3 main problems were observed:

- Project team members need a good knowledge about the tool and traceability in general to productively use TraceIt;
- In some project cases extra work was needed to keep up-to-date information in the database;
- A detailed analysis of methodology to be used has to be made prior the configuration of the model.

The above-mentioned problems are partly tied together. Ordinary project members do not like to think in terms of traceability. From the whole traceability model they more see the given item and some related ones. For tool, the ability to configure and the wide possibilities of applying different methodologies means there should be a person capable defining the process, the item types - not every project member can deal with this. Taking into account that many software development projects are with little initial knowledge of the process they would use, these projects rather would choose some predefined model and make the necessary adjustments. The idea is to provide some predefined models. At the same time, lots of projects do not have criteria for selecting a particular methodology.

It is little realistic that there could be a huge interest from industrial software development projects in a separate traceability tool. Real solution could be a high integration of traceability capabilities within other, more traditional tools like requirements management or problem tracking tools.

6. Related Work

Various research results reflect issues related to traceability models and traceability tools. At the same time, there are no explicit references available stating the traceability usage of the model as a basis for tool functionality. A comprehensive study on traceability issues is made by B.Ramesh and M.Jarke [15], including description of metamodels.

One has to mention that there are no unified principles of when exactly the term "traceability model" is used. For example, it may be also called less formally as "traceability relationships" [5]. Different approaches of describing the traceability model can be found, e.g. [16] proposes a comprehensive definition of the traceability model, but [11] describes model properties for UML-based projects. Issues related to the maintenance of the traceability model are outlined in [3]. Currently there has been no study of such aspect for the traceability model approach used in TraceIt tool.

Models provide an improved view on traceability and its evolution in [2]. But the provided traceability model there serves as an informative diagram without implementing it into some tool.

There are tools, like TOOR, made as part of the research projects that support definition of requirements, linking them to other project objects [13]. Some requirements engineering environments, like PRO-ART, are mainly oriented to pre-requirements traceability [14]. No explicit information is available of whether they are based on metamodel. Issues related to using a tool supporting the traceability are surveyed in [1]. Among central issues are data entry problem and granularity of information representation.

Commercial requirements management tools generally contain traceability as one of the features [10], and they are mainly tied with particular project life-cycle or methodology. Commercial solutions also contain integration between various tools, e.g. interface between requirements management tool DOORS and test management tool Mercury Interactive Test Director [18].

7. Conclusions and Future Work

This paper presents a metamodel-based traceability tool that was developed for an IT company. The distinctive feature of tool TraceIt is presence of the traceability model that: (a) defines the functionality of the tool, (b) is configurable at the project start-up and at later stages, (c) helps the project to organize the process.

The development of the tool was experimental, and it was a challenge to introduce a tool under the category "traceability tool". Many projects suspected this as a new burden for their work. The compromise was sought introducing it as problem reporting tool, or as an environment for defining and maintaining the tests.

Introduction of a new tool into project requires additional explanation of the methodology, the ability to improve solutions, the ways each particular function is realized. This could be a main difference from many academically developed tools that are finding little use in industrial projects.

Among observed benefits one could mention the ability to analyze the relations and dependencies between different project artifacts in several ways. In general, new projects are better suited to start using a new tool. Although it was evaluated that some of running projects could be suited of working with TraceIt tool as well, this was not done, as the process of switching from one approach to another could involve too many risks. Although it is well known that almost any software development process could benefit from the traceability information, it was discovered that the testing process up till now was one of the largest consumers of traceability information.

Among main observed problems one has to mention the additional qualification requirements for the project staff. There have to be people with a good understanding of the process, and because accurate maintenance of traceability links currently still requires some additional work. Although it was said that new the new tool should adapt to the project it always entails some changes into process, too. The goal is to make these changes in the direction of improvement.

Outlined benefits of the tool usage within a project could be considerably smaller if the projects would be using separate problem-tracking software. In this case, the information would be too scattered among various tools, and the optimization would require reducing the number of information systems used.

Prior, the IT company practiced traceability only via spreadsheets. In general, the attitude of the company towards tool improved, as projects were generally satisfied, even if they did not use all the intended features.

Open research question exists whether there is a place for software under the category "traceability tool". From user convenience point of view, there has to be a minimal set of tools to be used within every project. The traceability capabilities have to be integrated into other tools.

This paper outlined areas for further study. Among wide rage of issues, the future work can focus on optimal traceability models in the context of tool usage. Another issue is whether there could be considerably different requirements for large projects.

Integration with other systems could be reviewed. Main issue is the collection and synchronization of data originating outside TraceIt. Additional area for improvements could be related to further improvements in the implementation, including rising the level of usability.

References

[1] Arkley, Paul; Manson, Paul; Riddle, Steve. Position Paper: Enabling Traceability. 1st International Workshop on Traceability in Emerging Forms of Software Engineering, Edinburgh, UK. September 28th, 2002. p 5

[2] Baker, Loyd. Lessons Learned Applying Model-Based System Engineering Methods to a Strategic Planning Activity. Technical paper of Vitech Corporation. 1997, p 5. http://www.vitechcorp.com/infocenter/baker97a.pdf

[3] Bianchi, Alessandro; Visaggio, Giuseppe; Fasolino, Anna Rita. An Exploratory Case Study of the Maintenance Effectiveness of Traceability Models. 8th Int. Workshop on Program Comprehension (IWPC'00) 10 - 11 June 2000 Limerick, Ireland. pp. 149.

[4] British Standards Institution. The TickIT Guide Issue 5.0, 2001.

[5] Councill, Bill; Councill, Carol. Automating Requirements Traceability. STQE magazine July/August 2000, pp 49-54.

[6] Gills, M.; Bogdanovs, M. Extended Use of a Traceability Tool within Software Development Project.// H-M. Haav, A. Kalja (Eds), Databases and Information Systems II, Selected Papers from the Fifth International Baltic Conference, BalticDB&IS'2002, Kluwer Academic Publishers, 2002, pp 175-186.

[7] Gotel O.C.Z., Finkelstein A.C.W. An Analysis of the Requirements Traceability Problem. 1994. http://citeseer.nj.nec.com/78573.html

[8] IBM Rational RequisitePro http://www.rational.com/products/reqpro/index.jsp

[9] IEEE, J-STD-016-1995. Standard for Information Technology Software Life Cycle Processes. 1995.

[10] INCOSE, The International Council on Systems Engineering. Tools Survey: Requirements Management Tools. http://www.incose.org/tools/tooltax.html, 1998-2003.

[11] Letelier, Patricio. A Framework for Requirements Traceability in UML-based Projects. 1st International Workshop on Traceability in Emerging Forms of Software Engineering, Edinburgh, UK. September 28th, 2002. p 10

[12] Merant PVCS Tracker. http://www.merant.com/Products/ECM/tracker/

[13] Pinheiro F.A.C., Goguen J.A. An Object-Oriented Tool for Tracing Requirements. IEEE Software, March 1996, pp. 52-64.

[14] Pohl K. PRO-ART: Enabling Requirements Pre-Traceability. 1996 IEEE Proceedings of ICRE '96, pp. 76-84.

[15] Ramesh, Balasubramaniam; Jarke, Matthias. Toward Reference Models for Requirements Traceability. IEEE Transactions On Software Engineering. vol. 27, no. 1, January 2001. pp 58-93.

[16] Toranzo, Marco; Castro, Jaelson. A Comprehensive Traceability Model to Support the Design of Interactive Systems. 13th European Conference on Object-Oriented Programming, Workshop on Interactive System Design and Object Models. Lisbon, June 15, 1999.

[17] Telelogic. DOORS. www.telelogic.com/products/doorsers/doors

[18] TestDirector. Mercury Interactive. www.mercuryinteractive.com/products/testdirector/

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673. Datbases and Information Systems
220-231 P.

Topological Modeling and Arrow Diagram Logic Formalism Application for Software Development

Erika Asnina

Riga Technical University, Institute of Applied Computer Systems
1/3 Meza Street, Riga, Latvia
Erika.Asnina@dati.lv

Abstract. Problem domain analysis and modeling problems are discussed in the paper. The main problem of the object-oriented approach is its direction not to the problem domain, but to the application domain analysis and modeling. There is no formal connection between system functioning (dynamics) and a structure (static). Topological modeling is a convenient formal and comprehensive way to describe a problem domain in mathematical terms and to transform a system functioning model into a system structure model (class diagram). Sketch approach is a formal comprehensive way to precise a description of system class relations. The comparison of these two approaches and a small example of its application to the system modeling are given in the paper.

Keywords. Topological modeling, sketch approach, arrow diagram logic

1. Introduction

Adequate valid system modeling is the key for adequate valid design. It means it must be happened using some reasonable formal base [1].

There are two fundamental aspects to the modeling [2]: analysis, which defines what an application has to do with a problem domain to fit customer's requirements, and design, which defines how an application will be built. The line between analysis and design is fuzzy in the object-oriented development. "The problem domain is considered as a black box describing by use cases. Till now use-case driven object-oriented methods give low priority to problem domain modeling... They underestimate that only a proper problem-domain model provides a powerful language for expressing requirements to the system" [2], [3]. Hence, there is no formal connection between system functioning and a structure.

Modeling languages can be divided into the formal (as Z language), semiformal (as Unified Modeling Language UML) and informal in accordance with its' logical base. In practice, semiformal modeling languages are wide used, because its' using is easier, than formal languages using, and more formal, than in the case of informal languages.

A visual model described in some modeling language has to represent a real problem environment clear and valid. But there are some problems:

- *Semantic heterogeneity*. An amount of software kinds increases every year. This situation leads to the appearance of new problem domains, i.e. to different kinds of semantics [4].
- In order to deal with the heterogeneity one develops either new notational systems or extends existing systems (e.g. UML) [5]. It leads to the *syntactical overloading* of modeling languages.
- In the result of more complex requirements appearing, it is hard to separate specification aspects from realization aspects. Hence, specifications are mixed with realizations; i.e. semantic specifications often are weak. This problem is called by *semantic specification hiding in the realization.*
- Using of an inappropriate specification language or lack of an appropriate language is considered as the reason of the specification weakness. But every notational system is more or less straight representation of some formal logic. So, it means it is not a notational problem; it is the *specification logic* problem.

The topological modeling and the sketch approach can help to define a formal transformation from system functioning to a more adequate visual model of the system structure.

2. A brief description of the topological modeling

The topological modeling is formalism, based on assumption, that a complex system can be described in abstract terms as a topological space $(X< \Theta)$. X is a finite set of properties or functional features, and Θ is a topology in the form of a digraph (an oriented graph) [2], [3], [6].

A topological model of functioning can be constructed in the following sequence [6]: 1) to form a set X of essential physical or biological properties or features of the considered system. The word "essential" means "important for the normal functioning"; 2) to set a topology Θ in the form of an oriented graph or a matrix with indication of cause-and-effect relations among physical (or biological) properties or features of the system. Let's assume that there is a cause–and-effect relation between two properties, if one property appearance is caused by the other's appearance without participation of any intermediary property.

So, the sense of the analyzed system content is carried over abstract mathematical objects. Necessary knowledge for composition of the topological model of functioning can be obtained from the meaningful description of the system (in verbal, documented, analytical, statistical, etc. form) [2].

A topological model set three common topological functioning properties of systems:

1) Connectedness. Usually a system cooperates with other systems from the external environment. It means, the set X consists of two subsets M and N. M is the subset of elements describing functioning features of external systems, and N is the subset of only system internal features. Hence, there are not free (isolated) vertexes in the valid topological model of functioning.

It is possible to formulate the selection of a topological model of functioning from a topological space as the closure operation over the set N [2]. The finite

closed set of functional properties of a system is the union of adherent points of the set N, i.e. a point whose each "neighborhood" includes at least one point of N.

Such a topological model can be divided into subsystems in the same formal way, applying the closure operation to a subset of own subsystem properties. The formalized statements provide the control of correctness in the process of model construction [2].

2) Cycle structure. All functioning systems can be characterized by their cycle structure. We distinguish a main cycle and sub-cycles. The main cycle represents the main functioning of a system, but sub-cycles are main cycles of subsystems. A cycle is the feedback circuits in a system, as every element has to have a source and has to be the source for other element. Similarity of systems can be studied in details up to some predefined level [2].

3) Continuous mapping. Some functional property of a system can be considered as a more detailed subset of specialized properties. As it is stated in [2], if some more detailed functioning system is formed by substitution of a subset of specialized properties for some functional property, then continuous mapping exists between more detailed and simple parent topological models of the same system.

Corollary 1. In the topological digraph G* (X*, U*), the direction of arcs, which join the specialized point subset nodes with other nodes, is determinate by the direction of the arcs, which join the replaced point with the according nodes of the digraph G (X, U) [2].

Corollary 2. A data lack, which sometimes arises at the composing of a topological model of functioning, can be filled up by data that are obtained, when models of the same type systems are being continuously mapped into the model of the system under study [2].

So, the achieved model may be valid only in the case if the main cycle structure is found out.

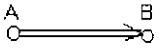
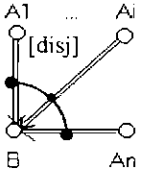
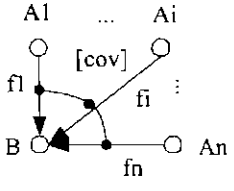
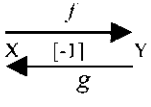
3. A brief description of the sketch approach

In the base of the sketch approach are principles of category logic [8] and arrow thinking [7], [9]. The basic idea consists in specifying any universe of discourse as a collection of objects and their morphisms. Objects have no internal structure: everything he/she wishes to say about them, he/she has to say in terms of arrows. It means an object structure and behavior are encapsulated and accessible only through the arrow interface. The distinction from functional data modeling is that here we deal with categories, not with pure sets, and constraints are hung on arrow diagrams, not on nodes. It allows dealing with semantic problems, mentioned in the introduction. More detailed about it is described in [7], [9].

In the sketch approach, a specification is an oriented graph, which consists of nodes and arrows with some marked fragments. These markers note predicates taken from some predefined signature. We call such a graphical construct by the sketch [8], [10], [11]. Sketches are noted as Π-sketches, where Π is the name of a diagram predicate signature. For example, UML language [12] defines a sketch signature Π_{UML} so that every UML diagram D could be represented as a special visualization of the logic specification S_D of the corresponding Π_{UML}-sketch. In common, any

diagram with precise semantics (described in mathematical terms) hides a sketch in some suitable signature of markers. Any given diagram property has a predefined shape, i.e. a configuration of nodes and arrows to which the property makes sense. A diagram predicate is specified with a name and a graph, i.e. the *logical arity shape* of the predicate. The arity shape should be supplied with auxiliary graphic means like arcs or double-body arrows for visualizing predicate declarations on schemes. In order to declare a predicate P with some arity shape G_P for a system S of sets and functions, he/she must assign S-sets to nodes in G_P and S-functions to arrows in G_P, in order that adjoinness conditions between nodes and arrows would be respected. It gives wide possibilities to any modeling language signature creating, and so to formal converting of diagrams into the sketch format and to sketches handling in the completely formal way. Table 1 shows few examples of predicates. They are set inclusion (Is_A), disjointness, covering, and inversion properties.

Table 1. Arrow diagram predicates for a system of sets and functions

Predicate name	Arity shape with visualization	Denotation semantics
Set Inclusion - the source set is a subset of the target set and mapping f is their inclusion.		$A \subset B$ *and* $f(a) = a$ *for all* $a \in A$
Disjointness - an element of the target set may be an element *of the only one* subset.		$\bigcup_{i=1}^{n} Ai \subset B$ and $\bigcap_{i=1}^{n} Ai = \varnothing$
Covering - each element of the target set is a value *of at least one* of the mappings f_i.		$(\forall b \in B)(\exists i \le n) b \in fi(A_i)$
Inversion		$(\forall y \in Y)\ \ f(g(y)) = y$

In order to handle sketches in a formal way, the category logic offers the so-called diagram operations over sketch [13]. Operations allow customizing sketches by extending them with derived items.

A diagram operation F is specified by a sketch D_F, denoting its interface in which a sub-sketch D_F^{in} of input data is designated, i.e. an operation F is specified by the inclusion $i_F : D_F^{in} \to D_F$ (D_F is an output sketch). The body of the operation is a procedure [[F]], which calculates an extension of derived items of D_F from the given extension of D_F^{in}.

The sketch approach supports basic modularization concepts of view and refinement of data. They can be described by *functors*, i.e. by arrows denoting a sketch mapping. The top-down analysis and design methodology suggests data schemes developing for a complex informational system in the process of stepwise data refinement. Each stage of refinement work can be specified as a mapping between sketches and the whole modeling process can be described by *hyper-sketches* [11]. Each arrow r_i is a pair (F_i, p_i), where the first element is a function F_i: $S_i \to S_{i+1}$ and the second element is a function assigning a refinement mapping p_i^S: $S \to \overline{F_i(S)}$ to any view scheme $S \in S_i$. The expression $\overline{F_i(S)}$ is the refinement of the view scheme S. This construction can be represented as a collection of hyper-sketches, each of them placed in its own fiber indexed by a number of refinement steps, where r_i' is inter-fiber mappings. The important thing is that this kind of diagram must hold commutative condition. It means that p_i is nothing but the so-called *natural transformation of functors*. Only if the commutative condition exists, the entire construction becomes a fibration in the technical categorical sense [8], [11], [13].

4. Comparison of formalization possibilities of the approaches

In order to use together the considered approaches, first we'll compare formalization possibilities of the approaches. An abstract topological model and a sketch is represented (logically and physically) as a graph or a graph-like structure $G(X, U)$, where U is a topology (a digraph) of X, which could be:

- A finite closed set in the case of the topological modeling. It is completely unimportant from the abstract viewpoint, what are elements of the set;
- Vertex sets, which can represent sets, object classes, data types, and theorems, and so on in the case of the arrow logic.

A topological model and a sketch satisfies the following requirements:

- A topological space must be connected (Isolated vertexes cannot be in it).
- Continuous mapping between simple parent and more detailed models of the same system.

So, the arrow logic satisfies and performs two topological modeling corollaries (section 2). An adequacy of a topological model and a sketch is achieved by assigning some proper meaning of the modeled system to the topological model or the sketch.

The universal arrow logic uses a predicate as a minimal logic unit. Such predicates define relations as well as operations. The sketch approach notation consists of tree units: a node, an arrow, and a predicate. All above-mentioned means arrow logic can represent complex data structures using fewer elements. But, there is one disadvantage in the arrow logic. There is no clear formalism for 1) system

topological model obtaining from environment topological space and 2) for obtaining a system subsystem. These two mechanisms allow investigating system differences and equivalence and evaluating importance and degree of differences. So, there is a lack of formal methods for system functioning modeling in the sketch way. System functioning can be represented by means of 2-arrow and 3-arrow graphs, which are difficult for comprehension [11].

So, *the abstract topological model* allows logically (but after some certain development also visually) representing any space topology and it is close to the sketch by the logical basis. In the both formats the main accent is set not on an object, but on its mapping into (or onto) other object.

But, is it possible and how is it possible to use together these two approaches in order to solve semantic modeling problems? First, we must not forget that the arrow diagram logic was "born" from the category theory [6]. TOP category, where objects are topological spaces and arrows are continuous functions between them, can be realized in terms of the category theory. In the paper [14] author describes the formal specification of the category TOP:

- An object in the TOP is a topological space (i.e. a topological model).
- An arrow between objects of the TOP category is a total function, which specifies equivalence between topological models' features.
- It can be traced how a particular property or a feature is decomposed into a set of properties and vice versa by the particular function mapping.
- Self-mapping will always have the same result: nothing is detailed or unified.

The topological modeling supports selection of system topological model of functioning from an environment topological space (a closure operation over the set N) [6]. This operation can be described in terms of the category theory. So, the relation (inclusion) between the category TOP and the category SET (with objects as sets) must be defined. The category theory allows defining a notion of a point. It sounds as follow — *a point of the object A is an arrow* $p: T \to A$ *from the category terminal object T to any object A of this category objects* [15]. A terminal (end) object is an image of all objects of the category. Otherwise, a mapping $p: S \to M$ from the unit set S to the set M in the category SET is considered as a point (an element) of the set M. The category theory allows defining the empty set that is an initial (elementary) object in the SET category. So, the closure operation over a set can be defined in the SET category framework. But selection of system and subsystems topological models in the TOP category is the subject of the future research work.

5. Formal modeling of the problem domain

There is an open question on the analysis and design process. Which approach is more preferable: to begin problem domain analysis with studying static data structure or dynamics of it? If one starts with use-case diagrams, i.e. with behavior, then he/she will have problems with static information diagram construction. On the other hand, beginning with static data structure analysis (class diagrams) requires knowledge about dynamics (associations).

A topological model of functioning and a sketch are the simplest way of formal mathematical describing of a problem domain. To prove it, let's consider a small example of business process modeling. Table 2 shows vertexes of the topological model of library work. Figure 1 shows the abstract topological model of the library. The model is very simplified; nevertheless, it captures the main features of the library functioning. The main properties of functioning are described with the topological model: *connectedness*, *a cycle structure* and *continuous mapping*.

Table 2. Functional features of the library

N.	Explanation	N.	Explanation
a_b	Selling of publications (external environment (EE)	h_b	Dividing of earned money
b_b	Registration of readers	i_b	Utilization of corrupted publications
c_b	Purchasing of publications	j_b	Financial support of state and private structures
d_b	Servicing of readers	k_b	Payment of employers work
e_b	Checking of publication condition	l_b	Informational support
f_b	Evaluation of reader requests	m_b	Publication of reports
g_b	Restoration of publications	n_b	Utilization company working (EE)
		o_b	People existing (EE)

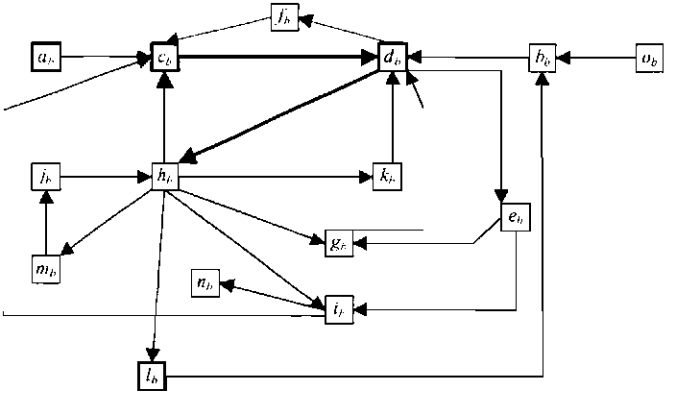


Figure 1. The abstract topological model of the library

It is important to emphasize that size and complexity of models aren't restricted in accordance with graph theoretical methods and tools [3].

The main cycle of functioning is ***c-d-h-c*** (symbols are without indexes *"b"*). There are eight sub-cycles in the model. 1) The ***c-d-f-c*** describes the subsystem that evaluates readers' requests. 2) The ***d-h-k-d*** describes the subsystem of financial support of the library employers' work. 3) The ***h-m-j-h*** describes the subsystem of financial support of the library work from the state and private structure side. 4) The ***d-h-g-d*** describes the financial support subsystem of publication restoration. 5) The ***c-d-h-i-c*** describes the financial support subsystem of publication utilization. 6) The ***c-d-e-i-c*** describes the process of publication utilization. 7) The ***d-e-g-d*** describes the subsystem of book condition evaluation and restoration. 8) The ***d-h-l-b-d*** describes the subsystem of financing of informational support. One can ask me why

I allot so many places for cycle describing. The answer is that a cycle is very important, it allows evaluating system functioning, defining crucial points in system functioning etc. [6]. Moreover, if one cuts up a system sub-cycle, the system will be malfunctioning, but if one cuts up the main cycle, the system will not be able to functioning.

As it was mentioned in the section 2, the topological model can be detailed thanks to the continuous mapping property. Figure 2 shows the topological model from Figure 1 in more detailed view. Table 3 shows vertexes, which extend the abstract topological model of the library. Additional information usually can be obtained from documentation about the system.

Table 3. Extending vertexes of the topological model of functioning

N.	Explanation	N.	Explanation
a_1	Book selling in EE	d_1	Book loaning
a_2	Magazine selling in EE	d_2	Book reading at the place
c_1	Book purchasing	d_3	Magazine reading at the place
c_2	Magazine purchasing	l_1	Information of the readers
		l_2	Advertising

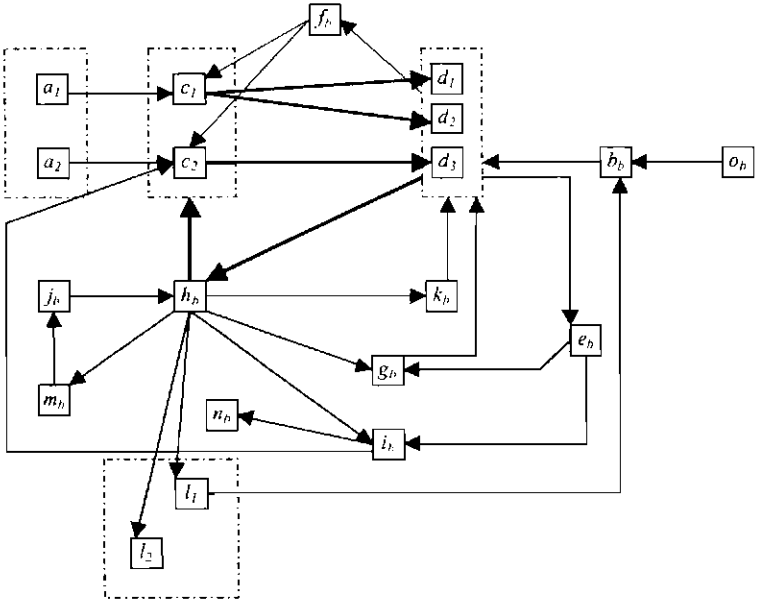


Figure 2. The detailed topological model of the system functioning

As you can see from Figure 2, the detailed model holds all features of the parent model, i.e. it is connected, with functional cycles and satisfies continuous mapping property. More about refinement of a system model is described in [3], [6]. Arrows going to or going out from a vertex block must be attitude to each of the vertexes in the block.

A topological model could serve as a formal basis for constructing UML diagrams. Such a model is an abstract and compact description that contents a lot of useful information about a problem domain [3]. Table 4 shows how vertexes represent concepts (conceptual classes). Figure 3 shows the conceptual diagram of the library with undefined relations between concepts.

Table 4. Concepts of the library system

N.	Explanation (*notation*)	N.	Explanation (*notation*)
1.	Publications (a_b)	13.	Analysis of reader' requests (f_b)
2.	Books (a_1)	14.	Restoration of books (g_b)
3.	Magazines (a_2)	15.	Division of earned money (h_b)
4.	Readers (b_b)	16.	Utilized publications (i_b)
5.	Purchased publications (c_b)	17.	Financial support (j_b)
6.	Purchased books (c_1)	18.	Employers (k_b)
7.	Purchased magazines (c_2)	19.	Messages to readers (l_1)
8.	Readers' services (d_b)	20.	Advertisement tasks (l_2)
9.	Book loan (d_1)	21.	Work result reports (m_b)
10.	Book reading at the place (d_2)	22.	Utilization company (n_b)
11.	Magazine reading at the place (d_3)	23.	Person (o_b)
12.	Evaluation of the publication state (e_b)		

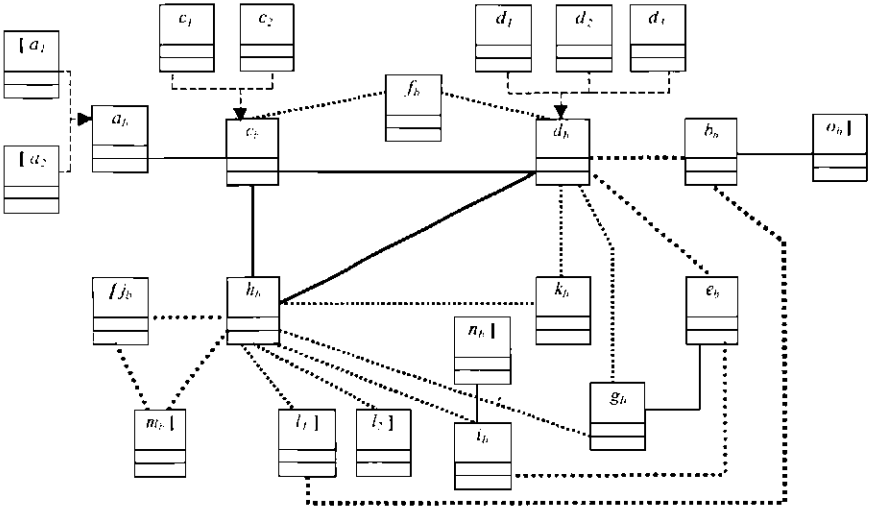


Figure 3. The conceptual class diagram of the library

I must emphasize, that *all vertexes (classes, concepts) of a conceptual class diagram must continuous map (one-to-one) into all vertexes of a topological model of functioning*. It is clear that now we can talk about information describing in a class diagram, for example an UML class diagram. The UML class diagram needs some improvement of the formal base. The following constructs can be embedded into the UML class diagram [3]:

- Input class: " | " before the class name;
- Output class: " | " after the class name;

- Inner class.

And the following (except generalization):

- Main cycle - ————
- Sub-cycle - ·····
- Generalization - - - - - - ▶

One of the convenient and formal ways for the UML formal basis improvement could be the sketch approach (section 3). This approach is similar to the topological modeling, and at the same time, it has more abstract and more expressive character. It allows formally analyzing relations between concepts in a problem domain at earlier stages of program development. Figure 4 shows how the topological model of functioning could be converted into the sketch. All vertexes of the sketch must be continuous mapped into the corresponding vertexes of the topological model.

Undefined relations between concepts could be represented as two inverse functions. All used arrow diagram predicates, except a *[graph]* predicate, are shown in Table 1. The predicate *[graph]* describes the relation, when source set elements are tuples of target sets' (vertexes') elements, i.e. m_i is (j_i, h_i) for all i (Figure 4). This predicate represents an association class of the UML language, but it shows such a property in more formal, comprehensive way. Such flexible features of the sketch approach as diagram predicates allow solving some of the specification weakness problems. A sketch is the expressive formal format for semantic modeling. Moreover, UML diagrams can be represented as sketch format abbreviations [16]. So, if some Π_{UML} signature would be defined for the UML language, then a sketch could be transformed into an UML class diagram in a formal way in accordance with the signature. At the end, I must emphasize that *each change in one format* (a sketch, a class diagram or a topological model) *must be noted in all other formats.*

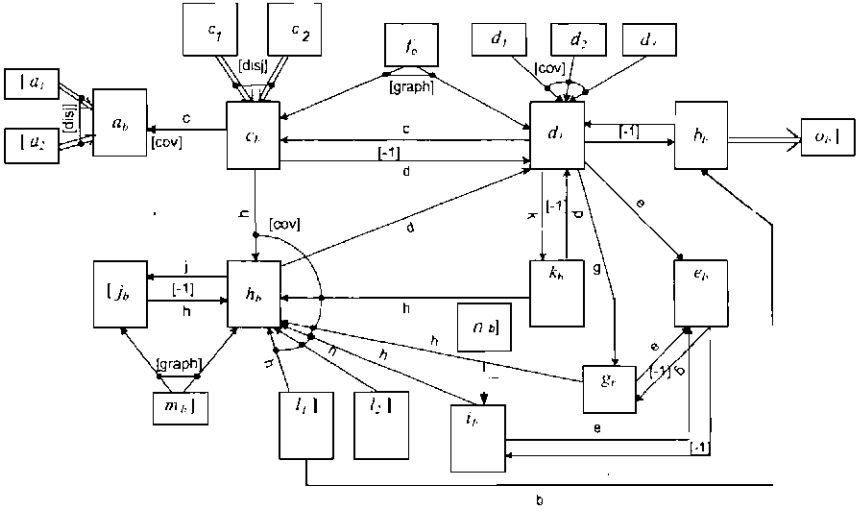


Figure 4. The sketch of the conceptual class diagram

Let's describe the suggested framework in common features (Figure 5). Fist of all, an analyst should analyze system functioning by means of the topological modeling

taking all the necessary knowledge from the available documentation. After that, the obtained topological model should be transformed into a conceptual class diagram. The conceptual class diagram defines the first assumption about existing concepts and relations. Then the conceptual class diagram can be transformed into a sketch. The sketch represents relations between problem domain concepts in the expressive formal comprehensive way. The sketch approach could be a notational system itself (and it has only three notational units: a node, an arrow and a predicate), or it could be a logical base of some other modeling language (e.g. UML). In the second case, the sketch should be transformed into a class diagram in accordance with the predefined signature of diagram predicates and operations over diagrams. Then the modeling language will be as some abbreviation of the sketch, but the sketch will be as the formal logical base of the system model.

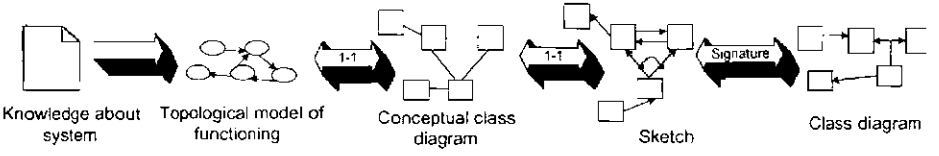


Figure 5. The suggested framework for problem domain analysis

Note there is an inclusion mapping (one-to-one) between a topological model of functioning and a conceptual class diagram and between a conceptual class diagram and a sketch.

6. Conclusion

There was considered two approaches possible usage in the system development process for the problem domain analysis. The formalization possibilities of the topological modeling and the arrow logic are very close.

The topological modeling offers a formal, compact and comprehensive way to transform problem domain functioning processes (dynamics) into structure (static). The topological modeling provides information mapping from the model of functioning to the class diagram. It provides formal changes management between system functional features and the according system structure. A cycle structure of the topological model makes it possible to evaluate system crucial points and thus decreases system development costs. The cycle structure allows system selection from the problem domain and subsystem selection from the system model. It solves the above-mentioned problem of *formal* component development.

The sketch approach and underlying arrow diagram logic offers the formal comprehensive format for complex relations among objects. It has small notation size (a node, an arrow, and a predicate) and is flexible and mathematically formal. The sketch approach could be used as the independent notational system and as the logical base of some informal or semiformal modeling language. It makes it possible to solve specification weakness and modeling language syntactical overloading problem as well as to answer continual complicating tasks.

But there are some disadvantages too. First, there is no mechanism for real-time system modeling in the topological modeling and the sketch approach. Second, the

word "formal" and especially words "mathematical formal" frighten program developers, because it requires large efforts already at the beginning of the development process. Of course, an appropriate modeling approach can be found for each task, but in the case of embedded systems or complex informational systems, the suggested framework can help developers to analyze and design systems more carefully.

References

[1] Osis J. Brief Survey of Object-Oriented Approach. Computer Science, Applied Computer Systems, Vol.3, Nr.2, Scientific Proceedings of Riga Technical University, Riga, 2001 - pp.23-32.

[2] Osis J. Topological functioning model support for software engineering. Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, 4th thematic issue, Riga, 2003 - pp.31-42.

[3] Osis J. Extension of Software Development Process for Mechatronic and Embedded Systems. Proceedings of the 32nd International Conference on Computers and Industrial Engineering. August 11 – 13, 2003,Limerick, Ireland, University of Limerick, pp. 305 – 310.

[4] Рамбо Дж. Interface Ltd.: Тенденция в развитии языка UML и разработке ПО. http://www.interface.ru/ (2001)

[5] Sinan Si Alhir. UML Extension Mechanisms. Distributed Computing, 1998, December - pp. 30-32.

[6] Осис Я.Я. Топологическая модель функционирования систем. Автоматика и вычислительная техника, Рига, "Зинатне", 1969, 44-50 с

[7] Asnina E., Osis J. Formalization Problems and Perspectives of the Program Development. Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, 3rd thematic issue, Riga, 2002, pp. 145-156.

[8] Barr M., Wells C. Category theory for computing science. 2. ed., London, Prentice Hall, 1995.

[9] Diskin Z., Kadish B., Piessens F., Johnson M. Universal Arrow Foundations for Visual Modeling. Proc. Diagramms'2000: 1st Int. Conference on the theory and application of diagrams, Springler LNAI, 2000, No.1889, pp. 345-360.

[10] Diskin Z. FIS/LDBD: Formalization of graphical schemas: General sketch-based logic vs. heuristic pictures. http://citeseer.nj.nec.com/diskin95formalization.htm, 1995

[11] Diskin Z., Kadish B. FIS/LDBD: The Arrow Manifesto: Towards software engineering based on comprehensible yet rigorous graphical specifications. http://citeseer.nj.nec.com/167037.html, 1998.

[12] J. Rumbaugh, I. Jacobson, and G. Booch. The Unified Modeling, Language Reference Manual, Addison-Wesley, 1999.

[13] Diskin Z. FIS/LDBD: Generalized Sketches as an Algebraic Graph-Based Framework for Semantic Modeling and Database Design. http://citeseer.nj.nec.com/diskin97generalized.htm

[14] Alksnis G., Osis J. Formalization of Software Engineering by Means of the Theory of Categories. Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, 3rd thematic issue, Riga, 2002, pp. 157-163

[15] Andrei Rodin. Endurance, Perdurance and Quantum Duality. Kaliningrad State University, http://philosophy.ru/rodin/EQ.htm

[16] Diskin Z. Mathematics of UML: Making the Odysseys of UML less dramatic. Practical Foundations of Business and System Specifications, 2003, pp. 145-157.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Databases and Information Systems
232 243 P.

Evaluation of real-time Data Warehousing processes

Janis Benefelds, Laila Niedrite

Department of computer science, University of Latvia
19 Raina boulevard, Riga, Latvia
Email: janis.benefelds@unibanka.lv, lnied@lanet.lv

Abstract. The focus of application of Data Warehouse with the lapse of time is changing from strategic planning and decision supporting to tactical day-to-day decision supporting and business process speeding-up issues. Therefore a concept of real-time Data Warehouse (RTDWH) becomes more popular. Classical understanding of the Data Warehouse is business data collection from many heterogeneous source systems, strategic analysis and decision support for high-level management. Real-time Data Warehouse tries to decrease gap or data propagation delay from operational systems to Data Warehouse to speed-up the getting of "right picture" and to lower the level of decision-making process in the hierarchy of organisational structure.
In this paper we introduce a way how to evaluate efficiency of the real-time Data Warehouse (assume a classical Data Warehouse is in place already) or in other words how to measure effort spent to implement real-time Data Warehousing solution by benefits, that come out from speeding-up and improving quality of business processes.

Keywords. Real-Time Data Warehouse (RTDWH), Data Integration, Business Process efficiency

1. Introduction

There was a time when transactional information systems started to grow up in terms of functionality and started to generate considerable volumes of information. Management started to feel a lack of information to make the right decisions how to run business. Classical Data Warehouse was introduced at this time. This solution is based on extracting business data from heterogeneous source systems, transformation and standardisation of them according to particular rules and uploading into the Data Warehouse. Data Warehouse (or Data Marts in advanced situations) ensures keeping of data in query-friendly format (dimensional data model, indexes, materialized views, etc.). Special data query, data analysis and data mining tools are available to operate with data provided by Data Warehouse. So, introduction of Data Warehouse solved a problem of having aggregated and summarised information covering all (or almost all) the business line of particular company. With the lapse of time the existing time period from entering data into transactional system and getting data from Data Warehouse didn't satisfy business users more and more. Depending from industry and even from different companies

within one industry the same time delay can or cannot be acceptable by business users. If one hour is acceptable for one, even one minute may be insufficient for another one.

Real-time Data Warehouse tends to decrease time delay in delivering relevant information to the key-actors within the business process. It helps then to speed-up response times between business activities and to improve quality of service level. One other result of real-time Data Warehouse is automated and rule-driven decision making process, which allows to avoid involving of high level managers, experts or key-actors to run relative simple data analysis and make simple decisions to pass it to the next business process.

"Real-time", "Near to Real-time", "Right-time", "Zero latency" and other very similar concepts are used when speaking about real-time Data Warehousing. The one thing, that is common for all those concepts, is "time". Since time (as variable) can be measured in particular units (day, hour, minute, second etc.), we've got a value – duration. Taking into account that in real world is no process, which takes time equal to zero, subject of discussion can be the value of time – duration. Respectively, is it acceptable (and therefore real-time) according to given business process or it is not.

There are many efforts to introduce the real-time Data Warehousing definitions [2, 10, 17]. In the scope of this paper by real-time Data Warehousing we understand such type of Data Warehouse architecture, which ensures delivering of relevant information to key-actors of business process according to their requirements within a time that satisfies business efficiency in particular business environment.

This paper introduces a method how to evaluate a real-time solution – is it worth wile to bring it in production in particular case or it is not. A metric like 'business value' together with time indication gives complete view whether the introduced solution gives some benefit or it doesn't. Using predefined restrictions and conditions we can measure and track the direction in which the particular real-time solution takes affect.

This paper is organised as follows. Section 1 is short introduction of Data Warehousing evolution to get a picture where a need for real-time comes from and illustration about what we really understand by concept "real-time". Section 2 shows the technological Data Warehousing solutions in data integration process and the possible implementations for real-time Data Warehousing. Section 3 conducts with evaluation of the real-time Data Warehouse and introduces a way to calculate some kind of efficiency to implement real-time Data Warehouse solution. Finally, section 4 gives conclusion of our paper.

2. Data Integration Process

The well-known definition of Inmon [7] states that a Data Warehouse is "subject-oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions". According to this definition one of important features of Data Warehousing is integration of different data sources into one consistent data store. Taking into account our real-time Data Warehousing

understanding we have to decrease data delivering time without affecting such Data Warehouse characteristics like data quality, data consistency, data volumes, response time etc.

The typical data integration process consists of capturing changes in data sources, extracting the changed data, transforming and loading them into Data Warehouse.

There are different ways of real-time data extraction from source systems [3, 4, 5 and 9].

In [13] there are given two possible data integration ways from data timeliness point of view. One is periodic integration, which means that data refreshments are done according to predefined time periods. Time periods usually are one day, one week or even one month. The second way of data integration is near real-time integration, which means that after appearing of new data within sources those data is immediately integrated into Data Warehouse.

Further there will be discussed the tasks, which have to be performed during the refreshment process, and the possible methods for implementation of these tasks according to two above mentioned data source integration ways.

2.1 Data source integration steps in case of predefined time periods

All integration tasks are described in detail by many authors, e.g. [2, 7, 8]. In [2] there is given a schema, which classifies the integration tasks in process layers, where different particular tasks are accomplished and then data is moved to next process layer (or step). The main data source integration steps according these layers are data extracting, data transformation and data storage.

The first step in data integration is capturing the changes in operational data sources. In periodic way of data integration the changed data can be collected and periodically extracted from data sources and moved to transformation process step. At present the batch processing approach to extract operational data is mostly used.

The data transformation step (cleansing, removing duplicates, standardising, calculating derivatives etc) makes the most complicated and time consuming part of all data integration process, also it has great impact on data quality in Data Warehouse. This step usually is performed in separate data staging area.

The last step in this chain of data integration processes is storage process. These tasks ensure actuality of database objects of the Data Warehouse, e.g. indexes, materialized views, and aggregates. In order to perform data storage tasks and data movement to the Data Warehouse, the periodical data integration have a basic assumption about existence of batch window – the time period when the access to Data Warehouse data by end-users should not be possible.

Data movement is transportation of all necessary data from one data integration step to the next one and it is done many times depending on how many steps do we have (see Figure 1).

2.2 Near real-time data source integration steps

The authors of [10] and [17] suggest necessary changes of Data Warehouse refreshment processes in the case of real-time data integration. Source systems are continuously monitored and data changes are captured and transformed as soon as they occur. The changes are moved into a Data Warehouse based on time periods as close to real-time as possible. In [12] the concept of continuous data flows is introduced, which means that the changed data are integrated into Data Warehouse permanently, in parallel with Data Warehouse users' analytical actions.

In the case of real-time Data Warehousing the data integration is based on processing of one changed data record, and the explicit data movement process is not necessary any more (see Figure 2). The changed data are directly integrated, e.g. inserted or updated into Data Warehouse schema using database transactions without additional data staging area data structures.

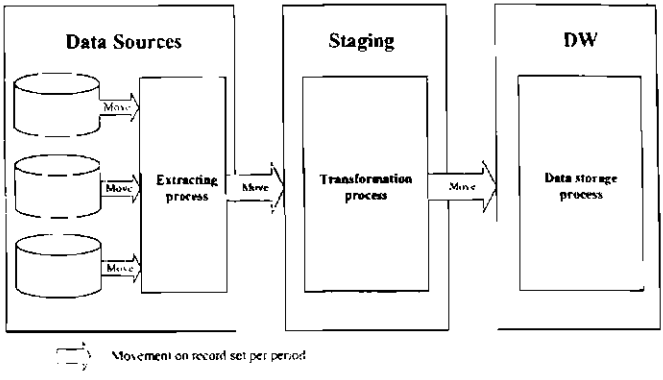


Figure 1. Data integration tasks in periodic data integration.

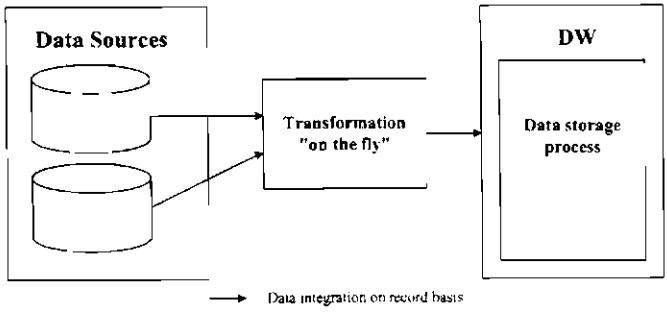


Figure 2. Data integration tasks in near real-time data integration.

2.3. Data source integration methods

2.3.1. ETL

Until recent the widely used method for data integration is Extract-Transformation-Load (ETL) described in detail in [6, 8] and many others sources. ETL is the process of gathering data from data sources, enhancing and enriching that data, and then loading it into a Data Warehouse.

The ETL process can be organized in different ways. For example, the approaches discussed in [1] and [16] focus on periodical data integration processes. Bouzeghoub et al. in [1] describe an approach, which is based on modelling the Data Warehouse Refreshment Process as a Workflow. Vassiliadis et al. describe in [16] an approach that also uses workflow solutions, but combines them with metamodel for data integration processes.

Traditional ETL tools are based on assumptions that they operate during a batch window and that there are no end-user activities during that time.

ETL is suitable in situations where the periodical data integration is possible and acceptable in time. The time periods for data processing can vary from weeks and days even until to minutes, which is close to real-time solutions, but even for short integration periods the key assumptions about batch processing remain unchanged. Some other data integration with ETL characteristics is given in [6]. Except time the amount of processed data and the complexity of integration and transformation processes can impact the choice to use exactly the traditional ETL solution.

2.3.2. Real-time data integration

For real-time Data Warehouse are two basic technologies for real-time data capturing, transformation and integration.

- The first technology uses the traditional batch processing as described earlier about ETL. The periods between two batch runs are minimized until hours or minutes. This implementation is called simulated real-time integration and has an advantage, because it uses existing ETL infrastructure with little changes. This technology is used when such delay is appropriate for Data Warehouse users.

- The second technology is continuous data integration. It uses messaging infrastructure.

Real-time data integration systems are based on messaging mechanisms. Particularly they are asynchronous and use publish-subscribe or event messaging architecture. These and many others data sharing and integration architectures are described in [5].

For the case of real-time data integration the most popular implementation is event messaging, which is more described in [4] and [10] as so called trickle feed,

and is based on the stack called message queue. Message senders, which in case of real-time Data Warehousing are data sources, put the messages into a queue. In event messaging is only one receiver e.g. Data Warehouse. The trickle feed mechanisms incorporate also data transformations in the message queue. The data load into the Data Warehouse in the real-time data integration process is not a separate step – data integration and end-user activities are not mutually excluded. That requires some specific database design approach to incorporate the delivered data records into historical data tables without any impact on end-user queries. One possible solution uses separate partition in Data Warehouse for new arriving data. This partition is not accessible for users. In every few minutes this partition is added to historical data table, and new partition for data feeds is created.

An alternative approach is based on the second messaging mechanism. It uses publish-subscribe architecture. One implementation of this architecture for real-time data integration is described in [12]. This implementation uses the concept of ETL containers for the staging purposes.

3. Real-Time's Data Warehousing Impact On Business Processes.

We stated already, that key issue is value of the time – duration, which can be evaluated as acceptable or not. If five minutes for somebody is acceptable, then even one minute for somebody is not acceptable – to long. Why? That is because everybody evaluates it against his own business processes. Why we have to evaluate real-time solutions against business processes? Because those are drivers of real-time Data Warehouse, thus those will be consumers of the real-time warehousing service. So, business processes serves as measure to evaluate is it worth to put an effort or not [11]. Beside there are a many other things that should be considered before starting a real-time Data Warehousing project [3].

Since Business Process can be as a measure we have to have some way, how to express real-time's efficiency by mentioned metrics. To describe that, we present a simplistic real world example from our observations. Assume there is a business process "P" (Figure 3), which consists of several steps (PN). A possible example of real world business process and its duration can be as follows (Table 1):

Table 1. Example of business process and its duration.

№	Description	Duration (min)
1	Customer calls to the bank and asks to score for his suitability to get a loan;	N/A
2	Officer asks for key figures and promises to "call back later";	2.0*
3	Officer enters key figures and orders "client scoring" in bank's loan application;	2.0*
4	Loan application transfers request to the Data Warehouse;	1.0*
5	When loan scoring is ready (2min), Data Warehouse informs the loan application (1min);	3.0*

Nº	Description	Duration (min)
6	Loan application lets to know the officer, that scoring has arrived;	0.5*
7	At the same time officer queries operational system for client's profile, because information from Data Warehouse will be one week old;	0.3
8	Officer calls to the customer and informs about information that was requested;	1.0*
	***Total of critical path**	**9.5**

To make this business process more real-time as it is in given example means decrease the total duration. Since it can be done affecting the critical path only, all activities, except activity number seven, should be in charge (Figure 3).

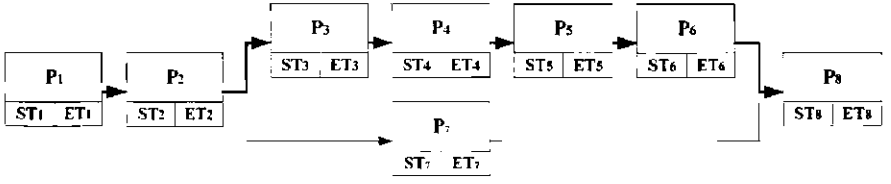


Figure 3. Example of business process PERT chart.

First of all, lets calculate the total time of business process:

$$T(P) = ET(PN) - ST(P1)$$

ET - End Time
ST - Start Time
T - Duration

Let as assume, we have managed to decrease the time needed for Data Warehouse to process particular data request by 1.5 mintues (to 0.5 minutes). We've got delta:

$$T_{\Delta} = T(P)_{BEFORE} - T(P)_{AFTER}$$

In our example it means about fifteen per cents of time.

How much is it? It could be evaluated as pretty good result in terms of time – officer can serve for fifteen per cent more customers than before.

Is our "new" business process real-time now? Answer could be: it is more real-time as it was before.

If we measure time delta by the business process it self, result isn't something sensational. Still the same number of activities, the same order of activities (no additional parallelism), the same input (data capturing in loan application and

client's profile request in operational system), the same deliverables (scoring result in loan application and client's profile in operational system), the same involved parties (officer, loan application, operational system, Data Warehouse). The only one benefit is time, nothing else. At this point we can define the following statement: until substitution of the classical Data Warehouse by the real-time Data Warehouse affects only time and doesn't affect other metrics of business process, it doesn't give very big benefit. Of course, everything is relative.

Usually data transportation from source systems to Data Warehouse and from Data Warehouse to Data Mart or even back to the source systems is based on ETL functionality. However more and more ETL functionality is combined with or substituted by real-time solutions (see Section 2) to improve communications between different enterprise applications and to decrease time that is needed for information transportation.

Let assume, that we have implemented cooperation between ETL and real-time data source integration technology to decrease significantly time, which was needed for data transportation between applications. It can be realised in a way, that customer gets his answer during the same telephone call (Table 2).

Table 2. Example of Business Process and its duration.

No	Description	Duration (min)
1	Customer calls to the bank and asks to score for his suitability to get a loan;	N/A
2	Officer asks for key figures and orders "client scoring" in bank's loan application;	2.0*
3	Loan application queries scoring data from Data Warehouse and client's profile from operational system;	1.0*
4	Officer informs customer about information that was requested;	1.0*
	***Total of critical path**	**4.0**

Since there is no activity parallelism this time, it is not needed to introduce PERT chart. Time saving can be calculated as it was done in the previous example. The most important thing is changed metrics of business process: fewer activities, less input (one data request only), less deliverable with the same quality (result is available in one screen), less involved parties (no operational systems in direct way). Now we can state, that new solution gives more benefit and savings. More real-time solution could be where customer serves him self.

Next step could be to introduce a web interface, where customer can get on-line answer to his request. Usually every financial company offers some kind of "loan facility" calculators, scoring instrument or similar tools. Most of them are based on data, that customer is capturing and changing every time running a scoring process. Of course, this is very good real-time service, but it doesn't have any affect on real business process. First, customer can do it in many other ways (other software, for instance) and, second, there are no real activities from company's side – customer

will have to go through the same process once more, when contacting company's officer. What we really have to introduce is facility to score customers capability based on real world data that are available in the company's Data Warehouse. Moreover, data must be "fresh" enough. What we get is: 1) customer can get information and apply for particular service whenever he wants (not only during office hours) and 2) there is no need for officer any more (customer serves him self on company's web site).

4. A REVIEW OF EVALUATION METHODOLOGY

Gartners research [15] has pointed out there are no reliable numbers to calculate the savings gains vs. the cost outlays to generate a financial ROI (Return Of Investments). However, we believe, the benefits can be grouped into two, business and technical, areas: 1) improved business operations and 2) lower costs of deployment, operations and support. Exactly improving in business processes is what we are focusing on in this paper.

The evaluation methodology, we would like to introduce, is based on the practical experience and research within Data Warehouse solutions in different type of companies and theoretical studies of business process management and assessment.

We introduce a metric called "Business value" here. We define it as a total of business process characteristics (for example investments needed, possible cost reduction, staff costs, product quality and/or flexibility, planed income etc.), scaled by its weight according to existing restrictions and priorities (for example limited financial resources, priorities, certain scope of employees or their skills, outsourcing possibility etc.). A general example how to calculate the business value is shown in the Table 4.

Table 4. Business value calculation.

Business Process Characteristic	**Weigh t**	**Rate**
BC_1	W_1	$BC_1 * W_1$
BC_2	W_2	$BC_2 * W_2$
...	...	...
BC_{N-1}	W_{N-1}	$BC_{N-1} * W_{N-1}$
BC_N	W_N	$BC_N * W_N$
	Business value:	$\sum BC_N * W_N$

As we are talking about real-time Data Warehousing, time as a value was analyzed separately from other business process characteristics. To evaluate a return of real-time Data Warehouse solution, two states should be evaluated – "input

business value" (starting characteristic) and "output business value" (resulting characteristic).

As we mentioned already, we do not include time within business process characteristics described above as it will be the second metric next to the business value.

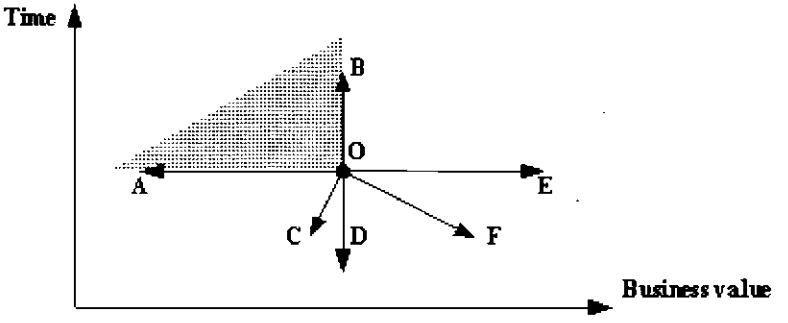


Figure 4. Real-time solution "Time-Business Value" matrix.

To evaluate the return of real-time Data Warehouse both input and output business values should be calculated and position them into the "Time-Business value" matrix (see Figure 4). Depending from the direction, in which the assessment has moved, correspondent conclusions can be drawn (see Table 5).

Table 5. Description of "Time-Business Value" matrix.

Direction	Description
A	No time changes and weaker business value than before. Certainly not the right choice.
B	Longer execution time and the same business value. Certainly not the right choice.
C	Shorter execution time but weaker business value. Should be evaluated very carefully but it's not the best practice. For example: we can save time on some data quality issues.
D	The same business value in shorter time. Very good start.
E	Better business value in the same execution time. We've got better business solution but nothing new regarding real-time issues.
F	Shorter time and better business value. Both characteristics have improved. Right direction to continue.

To comment "Time-Business value" matrix in more details we have to look at the different areas. Area AOB is the worst direction at all. Theoretically it is possible, that improvement efforts go this direction, but there should be some mistake in calculations then. Physically we can't imagine such kind of situation. Area BOE (together with AOB) describes performance decreasing (longer execution time than before). Since we are going to evaluate real-time solution, this wouldn't

be the right direction too. Are AOD satisfies time decreasing requirement but decreases business value metric. This is a tricky one situation. If the "improved" version of Data Warehousing solution is evaluated to go this direction, decision to do or not do should be evaluated really very carefully – characteristics, which decreases business value should be pointed out and analysed in more details. Sometimes it is worth to improve performance at business value cost, but usually it is not. Finally area DOE – this is the right direction to go. Improving both, time and business value, is the main goal of introducing real-time Data Warehouse solution.

It seems there will be very big emphasis on having real-time data not only for decision-making as well as for business process optimisation and improving operational performance [14].

5. Conclusions

As conclusion of our paper we would like to emphasize that meaning of the term "real-time" can be even very different for particular real-world situations and it can change (and usually it does) in the lapse of time. By different real-world situations we have different type of business processes with different type of restrictions and priorities. It is very critical to have real-time Data Warehousing in financial companies, healthcare institutions, and communication entities. Real-time in, for example, education is not understood by seconds or minutes. Usually it is enough to have information on the weekly or even monthly basis. Mentioned examples vary from the restrictions' point of view – healthcare and education institutions usually don't have so much free financial resources as financial or communication companies do.

The main issue we would like to emphasize, is real-time Data Warehouse solution dependency on characteristics of each particular case (restrictions and priorities). Of course, there are some general guidelines that should be followed any way.

We evaluated two source data integration methods in association with different restrictions and benefits (parallel data refreshment and end-user activities vs. mutually excluded activities; data refreshing on the basis of predefined time periods vs. data adding to Data Warehouse as soon as data are changed within the source systems). Each of the methods should be used where it gives the most benefit (business value). It is not reasonable to implement an expensive real-time Data Warehouse solution if it doesn't improve business process or there even is no need to change them. To avoid such type of mistakes we have introduced one way how to evaluate forthcoming (or offered by somebody) real-time solution. Such "Time-Business Value" evaluation matrix is applicable in the process of choosing between several possible real-time Data Warehouse implementation scenarios as well. So, you just have to choose within a matrix the right direction you wish or are able to go.

References

[1] Bouzeghoub, M., Fabret, E., Matulovic-Broque, M., 1999. Modeling the Data Warehouse Refreshment Process as a Workflow Application. In Proc. of DMDW'99, Heidelberg, Germany.

[2] Bruckner, R.M., List, B., and Schiefer, J., 2002. Striving towards near real-time data integration for Data Warehouses. LNCS 2454, pp. 317-326, Springer Verlag, Berlin, Heidelberg

[3] Fuller, R.D., 2003. The Fundamentals of Data Warehousing: Real-Time Data Warehousing, May 27, 2003, www.datawarehouse.com

[4] Godadia, R., 2004. Right in Time, Intelligent Enterprise, February 7, 2004, www.intelligententerprise.com

[5] Haughey, T., 2003. Data Integration And Sharing, The Data Administration Newsletter, Oct., 2003, www.tdan.com

[6] Haughey, T., 2004. Data integration and sharing - part two, Data Administration Newsletter, Jan., 2004., www.tdan.com

[7] Inmon, W.H., 1996. Building the Data Warehouse. Second Edition, John Wiley & Sons, New York.

[8] Kimball, R., Ross, M., 2002. The Data Warehouse Toolkit, John Wiley & Sons, 2nd edition.

[9] Orr, K., Cutter Business Technology Council, 2002, Building a Real-Time Enterprise: Why It's Worth the Effort, Enterprise Architecture Advisory Service executive report, Vol.5, No.10

[10] Raden, N., 2003. Real Time: Get Real, Part II , Start by discarding your current concepts of ETL , Intelligent Enterprise Magazine, http://www.intelligententerprise.com, June 2003

[11] Ramankutty, P., 2003. Is "Fully Integrated" Really the Best Thing?. Business Integration Journal, Sept., 2003

[12] Schiefer, J., Jeng, J.J., Bruckner, R.M., 2003. Managing Continuous Data Integration Flows, In Proc. of the 15th Conference On Advanced Information Systems Engineering (CAiSE'03), Springer LNCS, Velden, Austria

[13] Schwinn, A., Schelp.J., 2003.Data integration Patterns, Proceedings of BIS 2003, Colorado Springs, USA

[14] Smith, M., 2003, Operational Performance Management, Ventana Research, Dec 11, 2003, Article ID: V03-25, www.ventanaresearch.com

[15] Strange, K., 2002. Data Warehouse Scenario: TCO and ROI in Tough Economic Times. Gartner Symposium ITxpo, San Diego, 29 April – 2 May, 2002

[16] Vassiliadis, P., Quix, C., Vassiliou, Y., Jarke, M., 2001. Data Warehouse Process Management. Information Systems, Elsevier Science, Vol. 26(3): 205–236.

[17] White, C.,2002. Intelligent Business Strategies: Real Time Data Warehousing Heats Up, DM Review, August 2002.

SCIENTIFIC PAPERS UNIVERSITY OF LATVIA. 2004. VOLUME 673, Datbases and Information Systems
244 250 P.

Availability of Database Services in Estonia: How X-road is Progressing.

Kristiina Kindel

Institute of Cybernetics at Tallinn University of Technology
Akadeemia tee 21 12618, Tallinn, Estonia
kristina@cs.ioc.ee

Abstract. Estonian state databases have been arranged since nineties as the state has duty to gather quality information and to allow access to that. For easing interchanging data between databases and allowing people to get their data easily the Ministry of Transport and Communication of Estonia has launched the project X-road. This paper presents databases services in Estonia and describes how X-road has progressed so far.

1. Introduction

Systematized information is the bases of establishment's functioning. Thus the state databases are very important data sources. In Estonia, as in most other countries, so far databases were relatively standalone and were supposed to gather information about some certain objects like people, cars, houses, land and ships.

As one principle of information society is to guarantee free information flow, then it is extremely important to forward data that is gathered and managed by state to those, who need it. Solving various information technology tasks becomes important to get information from several databases simultaneously. Easiest way to solve this problem is to get data over the Internet. Accessibility to data in databases should be service-centred.

Provided data must not violate principles of state defence or personal data protection. Person who needs information might be citizen, civil servant or proprietor.

X-road is created for solving service-centred data gathering problems. There has been opened several services already and the purpose of this paper is to analyse these services and describe the project X-road in Estonia. The paper is organised as follows: section 1 gives short overview about X-road, section 2 is about government-to-government services and in last section are described public services.

Service-centred architecture is main solution for modern distributed information systems.

2. X-road

The aim of X-road project is to develop software, hardware and organizational methods for standardized usage of national database and information systems [3].

X-road is the modernization program of national databases with the aim to change national databases into a common public, service-rendering resource, which would enable agencies, legal and natural persons to search data from national databases over the Internet, provided they are entitled to do so. At the same time, the system will ensure sufficient security for the treatment of inquiries made to databases and responses received. [7]

Using registers via X-road started at the same day when Estonian citizens ID-card was put into service, 28 of January 2002. Soon banks joined with X-road for offering authentication service. After launching the X-road citizen's portal enterprise's information systems and databases started to join. Some of them give as well as get data using X-road.

Development of X-road has been extended already two years; lots of Estonian databases and information systems have been joined. Similar to the citizens' portal there is another version named MISP (Mini Information System Portal) witch allows the civil servant institutions to use X-road in the same way as citizen uses citizen's portal. Compared to citizen's portal MISP has several differences. One of them is that every civil servant will be additionally authorized besides authentication. It means that every servant gets rights to see only such data that is related to his/her duty. MISP is not technically mandatory for enterprises joined to X-road if they have their own information system that consists opportunity to authentication and authorization of employers. In this case it is possible to use X-road queries through enterprise information system.

For today there are already over 50 organizations joined to X-road. By statistics, most enquiries are made by police, near 4000 queries per day.

3. Government to government services (g2g)

There are about hundred databases registered in State register of databases (ARR). The usage activity and data capacity of them are very different. There are tens of databases witch are functioning 24*7. X-road is joining mostly of them. Some of those databases have web-services but X-road's advantage is that users don't have to remember all the passwords for different databases and don't have to make a data usage contract with each of them. [1]

For a public institution there are services that the institution needs for its everyday work according to access rights to the data. For example, notary's enquiry about citizen's vehicles from cars database, or about citizen's enterprises from business register. Or police enquiry about driving licenses of citizens. Public institutions have similar user interfaces than citizens; only the list of services differs according to their access rights.

Presently are joining X-road following registers: register of second column of pension; Department of the Treasury information system; Board of Boarder Guard

information system; Customs information system; Labour Inspectorate information system; state register of work seekers and work market services; security centre databases; State Register of Cultural Monuments; State register of railway rolling stock; State railway register and Register of the Estonian diplomatic passports. [8]

To enhance g2g services X-road has ordered some expansions. First and larger expansion was reorganising ARR so that it will become as part of X-road's working environment. New ARR has the same main functions like central registering databases and being information-gathering environment but attached some functions that make it more service-based environment. New ARR will be the tool for every X-road user. There will be described Estonian databases e-services and their usability. Every organisation that wants services from some database looks first weather there is already some similar service for other organisations in ARR. If there is such service and the organisation needs that data for its everyday work and has right to see that then this service will be opened for them also. If not, then wish of service will be registered and database developers will realise it. Second expansion came from need to carry over large amount of data through X-road. This need came firstly by solving State court adjudication register and court solutions of first and second instances of Ministry of Justice replication mission using X-road. Third large expansion was to support mutual realisation of services between databases of Citizenship and Migration Board and databases of population Register. This was very important for modifying data quality of two large registers that reflect personal data. [8]

4. Public services

By law citizens have the right to see the data government has about them in state databases. Before X-road you had to go to the accredited processor, write an application and wait some days and you got your information. Now it is easier, you need only computer with internet connection and with browser and you can ask about your driving licenses, passport data, lands, vehicles and so on. The answers are with few second on your screen.

For now there are about 25 services for citizens [5]. The main registers that offer services are as follows: traffic register, business register, population register, land register, ships register, Citizenship and Migration Board register and pension insurance register.

There are two ways to identify person. One way is through banks. For that person must have been joined to internet-bank. Entering to the citizens portal, person gets fist the list of banks through witch he/she can identify him/herself. User chooses the bank where he/she enters identifications and passwords. After that person is directed back to the citizens portal and bank gives to portal persons personal code. The other way is to use ID-card. Currently, internet-bank is mostly used for identification, because there are more internet-bank users than ID-card owners. Although there are already over 350000 ID-card owners in Estonia. Presumably it's going to rise more rapidly near future because of Tallinn's new public transport tickets system, which uses ID-card to offer Tallinn people more reasonable prices than others. So it is useful to own ID-card.

Citizens can see the data only about themselves. Citizen's portal allows to everyone who has Estonian personal code to make those queries. Every person can see only the data about him or herself, for example what car he/she owns, what kind of driving licenses he/she has and so on. Because of opportunity to see their own data, lots of people have discovered some mistakes in their data, mostly in population register. For that reason some registers have created services that allow people to denote or correct the mistakes in their data.

5. List of services

Subsequently are described services in citizens' portal. In Fig. 1 is depicted fragment of citizens' portal. Services are in Estonian language and listed by registers.

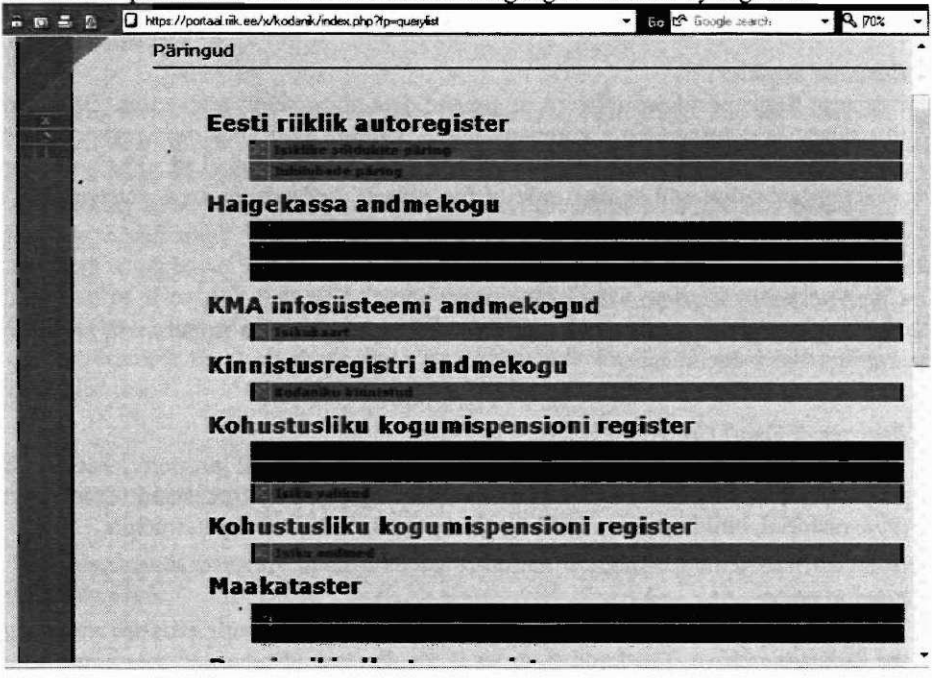


Fig. 1. Citizens' portal

Estonian Traffic Register

There are two services: First one is about person's vehicle(s) witch gives car's registration number, mark, color, building year, motor's power, is it dispossessed or not and owners address and name. Second, driver's license query gives persons name birth date, personal code, place of residence, drivers license number, type, stipulations, period of validity and status (is it valid or not).

Population Register

Population Register started from a need to compose the list of citizens for monetary reform. Very shortly after gaining independence Estonian government decided to introduce its own money – the Estonian crown instead the Soviet ruble. This operation needed to have an exact list – who are the citizens of the country and who has the right to exchange the money [3].

Population Register offers three services plus one temporary service that is added before the elections, so the citizen can control that he/she is in the voting list.

First service shows persons name, gender, birth date, address, documentation data, nationality, mother language, and education. Other two services are for denoting mistakes in person's data. In first one, you insert your personal code, first and last name and in free form the description of the mistake. The second service allows citizens to change their statistical data like nationality, mother tongue, education, and activities. It is possible only once in 30 days or 3 times in year.

Business Register

Business Register allows citizens to see the data about their enterprises. Output of this query is entrepreneur's company name, validity time, business register code, number of enterprises connected to that person, legal status (Ltd., PLC, not-for-profit organization and so on), role of the person, registration area, entrepreneur's status, communications and so on.

Citizenship and Migration Board Register

That register gives persons personal data, picture that is also in passport, signature and data about documents.

Register of Small Crafts

Gives person's data; data of craft's property like kind of property, section of property, date; ship data, registration number, registration time, name, home port, type, material, building year, building place, constructor and superstructure.

Land Register

The Land Cadastre consists of the Land Register together with cadastre maps and the cadastre archive. The Land Register is a collection of recorded data concerning cadastre units. Query is made by cadastral unit and gives following data: cadastral identification, official name of cadastral unit, location, code of county, self-government's code, registration date, technical codes of intended purpose, per cents of intended purpose, area, coordinates and so on. Second query is by location.

National Pension Insurance Register

This register offers three services for now. First one shows how much social taxes person has paid thorough years and second one lists out pensions and supports that person has. Third one and the newest allows to citizen to make an application to get a parental compensation without going out of home. This service gathers information also from health insurance register, population register and department of the treasury.

Health Insurance Register
Health insurance register shows to the citizen three queries: Personal data, Health Insurance and Financial Compensation. First query shows citizens personal data that is in that register like name, personal code, gender, birth date, address. Also allows making corrections in your data if something is wrong. Second query shows your health card number, personal doctors name, in what area doctor works, kind of insurance, how long it lasts, employer who pays the health insurance. Third query shows compensations that person has had and allows making an application to extra compensation of medicals.

Bailiff database.
Shows whether there is some enforcement procedures initiated against citizen.

6. Summary

X-road is for connecting different state databases to each other. For the conclusion, there are lots of services working now and the number of them is growing all the time. X-road has spread as extensively as planned beginning of the project. Availability of services is improving every day, the number of services is enlarging and more and more registers are joining with X-road. However there are some registers witch haven't joined because of that register is not functioning properly jet. The aim of X-road is that all transactions with state are realised without going in place, it also helps to improve the usability of ID-card. Joining with European Union our government must open access to state databases for EU. X-road technology is ready for that.

7. Acknowledgement

This research is partially sponsored by Estonian Scientific Foundation under the grant nr. 5766.

8. References

1. Development of X-road. http://x-tee.riik.ee
2. Kindel, K. Estonian Largest Databases: State of the Art and Services. B. Thalheim, Gunar Fiedler (Eds), Emerging Database Research in East Europe, Proceedings of the Pre-conference Workshop of VLDB 2003, Computer Science Reports, Brandenburg University of Technology at Cottbus, 2003, 14/3, pp 74-76
3. A. Kalja. System Integration Process of Government Information Systems. The Proceedings of PICMET'03, Portland, Oregon USA, 20-24 July 2003, 6 p.
4. State Register of Databases. (Andmekogude Riiklik Register). http://www.riik.ee/arr.
5. Services http://x-tee.riik.ee/eteenused
6. Citizen's portal. http://www.riik.ee/kodanikuportaal/
7. RISO http://www.riso.ee/en/index.html

8. Kalja, A. X-road's data traffic is open second year (X-tee andmeliiklus on avatud teist aastat). IT in Public Administration of Estonia. Yearbook 2003 (IT avalikus halduses. aastaraamat 2003)