UDC 004.5:519.876.2

M. Iwaniak, W. Khadzhynov
Technical University of Koszalin
Department of Electronics & Informatics
ul. Śniadeckich 2, 75-453 Koszalin, Polska
hadginov@ie.tu.koszalin.pl

Coloured Petri Net Model of Two-Phase Commit Protocol With Multiple Participants

In this work the usage of Coloured Petri Net for modeling and simulation of Two Phase Commit protocol with multiple cohorts is presented. Brief overview of Coloured Petri Nets is introduced. Two-Phase Commit protocol is briefly described than introduced as Coloured Petri Net model. By initial marking adaptation a few scenarios for 2PC protocol are presented with the use of the reachability analysis.

***Key words**: Coloured Petri Net, distributed transactions, 2PC, Two-Phase Commit protocol.*

1. Introduction

The theory and the practice of centralized database system is well developed. During many years of studies the standard that would be helpful in projecting and implementation of systems based on distributed database was not developed. There is especially a lack of operative standard for heterogonous distributed database.

The theory of Petri Net is used in modeling and analyzing parallel processes. The structure and function of Petri Net may be described in algebraic form and processed with the use of numerical methods. The basic principles of how Petri Net works are converge with the essence of transaction processing. In both cases of Petri Net performance and transaction commitment are possible only after fulfilling all the necessary conditions. During research work these similarities may allow us to create a new specialized modeling language.

Our earlier studies presented in [1] showed the Petri Net place/transition net model of classic Three Phase Commit (3PC) protocol. A number of limitations and problems were pointed and applying Coloured Petri Net was suggested.

In this work we present Coloured Petri Net model of Two-Phase Commit protocol with two participants. In section 2 fundamental information of Coloured Petri Net is described. Section 3 describes Two Phase Commit protocol.

2. Coloured Petri Net

Most commonly described is The Ordinary Petri Network, also named the network of class position/transition. Coloured Petri Net is classified as a high-level net. The introduction of different types of tokens is the main extension in this class of Petri Net. It's graphical representation is bipartite graph which contains two types of nodes connected by arcs. Nodes of the net are:

— places — represented by circles;

— transitions — represented by rectangles.

Petri Net can be expressed as the triplet $N = (P, T, D)$ where:

— P is a collection of places $|P| = m$;

— T is a collection of transitions $|T| = n$;

— $D = [D^+ - D^-]$ is incidence matrix of dimension $m \times n$, this matrix is created by subtracting pre-incidence matrix from post-incidence matrix, it contains elements $d_{ij} = d_{ij}^+ - d_{ij}^-$. These elements come from subtracting weights of input arc from weights of output arcs;

— D^- is pre-incidence matrix of dimension $m \times n$, it contains elements $d_{ij}^- = w(i,j)$ which describe the weight of transition j input arc, that is the arc directly connecting place i with transition j ($P \times T \to N$);

— D^+ is post-incidence matrix of dimension $m \times n$, it contains elements $d_{ij}^+ = w(i,j)$ which describe the weight of transition j out arc, that is the arc directly connecting transition j with place i with ($P \times T \to N$). (Все абсолютно правильно)

In Coloured Petri Net for the D^+, D^-, D matrixes the weight represented by algebraic notation representing each color tokens quantity.

2.1. Markings of Coloured Petri Net and transitions firing

Marked Petri Net can be described by four-tuple $PN = (P, T, D, M_0)$, where M_0: $P \to \{0,1,2,\ldots\}$ is an initial marking of the net that defines the token distribution in the network places. In the Coloured Petri Net we define different classes of tokens. These classes are called colors.

For a specific marking there can occur dynamic events in Petri Net. If the input places of given transition have an amount of tokens equal to input arc weight then the transition comes into enabled state which means it can be fired. After firing of the transition from all input places the tokens are taken away and new tokens are inserted into the output places. The amounts of tokens taken away from each input place and inserted into output places are equal to weights of given input and output arcs. The process is presented on Fig. 1.

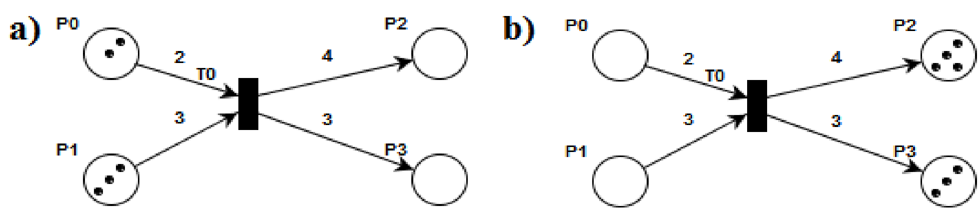

Fig. 1. Transition: a) transition ready to be fired; b) transition after firing

In Coloured Petri Net places can hold tokens of different colors at the same time, arc weight can be described by any of defined colors.

2.2. Reachablity analysis

In the reachablity analysis we study subsequent marking changes that happen after transitions firing for assumed initial marking. Typically reachability analysis is presented by reachability tree which is a directed graph build with such elements as:

— the root — representing initial marking;
— nodes — representing each reachable marking;
— arcs — representing fired transition,
— leaves — representing final or repeated markings.

We begin reachability analysis by checking enable/ready transition for assumed initial marking. After transition has been fired, a new marking is created. Determination of another marking can be simplified with usage of incidence matrix *D* which describes our Petri Net structure.

Each column of the incidence matrix describes a change in marking after a firing of given transition. Each marking can be computed with usage of the following formula

$$M_k = M_{k-1} + e[t_j]\, D,$$

where $k = 1,2,3,\ldots$ and $e[t_j]$ a firing vector containing a digit 1 in position corresponding *j* transition.

3. Two-Phase commit protocol for distributed transactions

To execute transaction in distributed database system the additional communication is needed. This communication occurs between Transaction Manager (TM) which acts the role of transaction coordinator and other TMs that act the role of participant (sometimes called cohorts). The way of communication between TMs is defined by distributed transaction commit protocol. Most widely used and featured is Two-Phase Commit protocol.

By its name 2PC protocol realize two phases: phase of voting and phase of commitment. Voting phase takes place after Coordinator sends initial message to Cohorts asking if they are ready to commit distributed transaction. Readiness to commit is send back by Cohorts. After Coordinator collects returning messages from all Participants commit phase begins. If all collected votes are *vote-commit*, Coordinator sends message of *global-commit.* If any of Participants sends *vote-abort* or does not respond at all Coordinator sends message of *global-abort*. Making of the decision only when all of the votes are *vote-commit* is the main way for assure atomicity of distributed transaction.

On Fig. 2 the algorithm of 2PC protocol is presented. Fig. 2 was prepared upon [1]. The circles represent states of Coordinator and Cohort processes. Rectangles represent operations of logging received messages and made decisions into system log. Arrows represent the flow of messages and control.

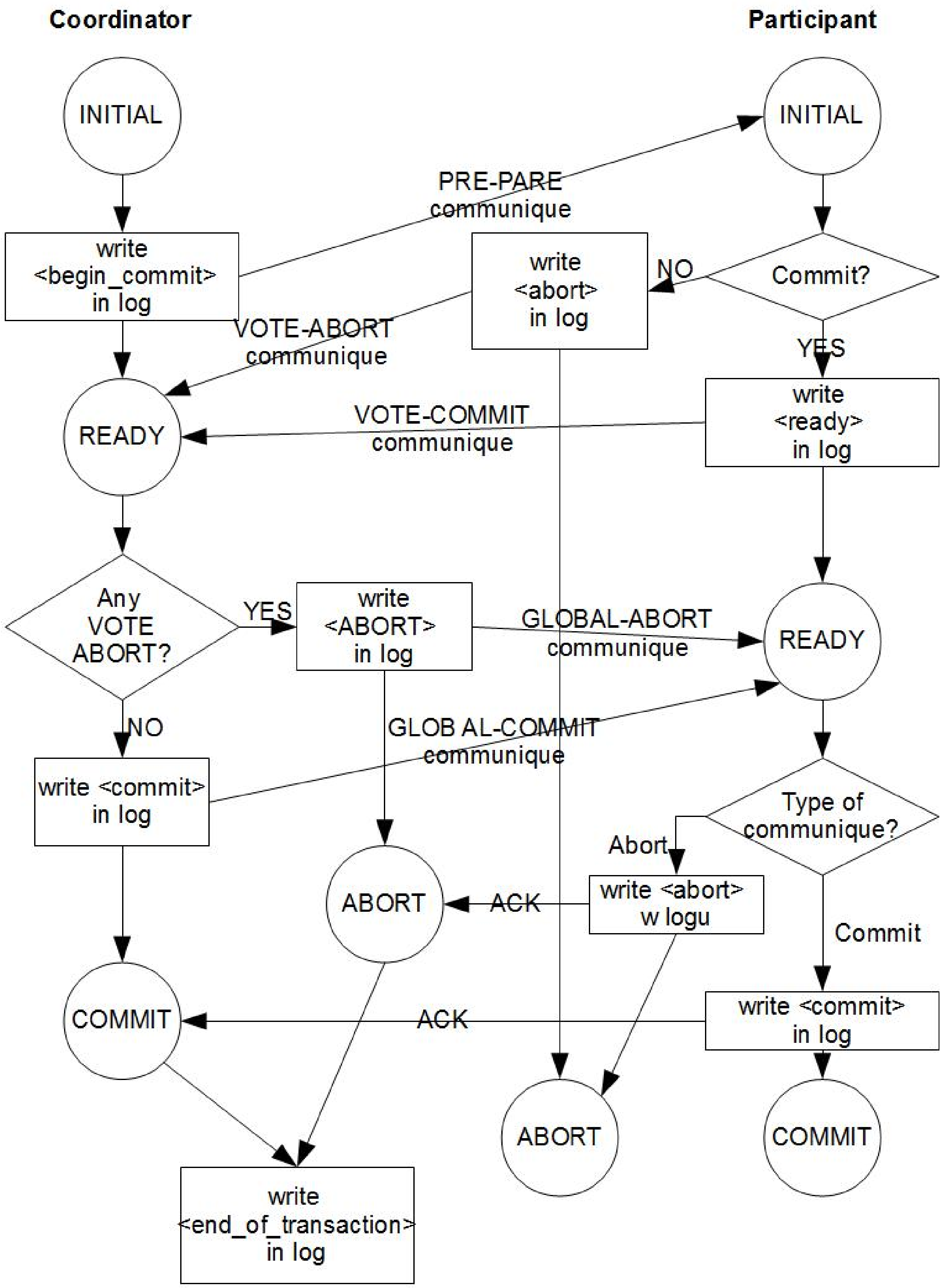

Fig. 2. Two Phase Commit protocol algorithm with one participant

4. Coloured Petri Net model of Two-Phase Commit protocol with two participants

Fig. 3 presents Coloured Petri Net model of two participants 2PC protocol implementation. Places located near right edge of figure shows first participant (U1) while places located near left edge of figure shows second participant (U2). Coordinator places are located in the center of Fig. 3.

In proposed Petri Net model following token colors were introduced:

— INIT (I) — refers to initial communique;

— COMMIT (C) — refers to positive messages like vote-commit, global-commit or commit

— ABORT (A) — refers to negative messages like vote-abort, global-abort or abort.

For a given arc weights of colors can be defined. In graphical presentation of Coloured Petri Net commonly different weights are placed near the arch and have differ colors. Colors used on Fig. 3 may be indistinguishable in print, accurate weights of used colors were placed in Tables 3 and 4.

Table 1 presents places (states) and transitions (events) representing coordinator process.

Table 2 presents places (states) and transitions (events) representing cohort process upon cohort U1.

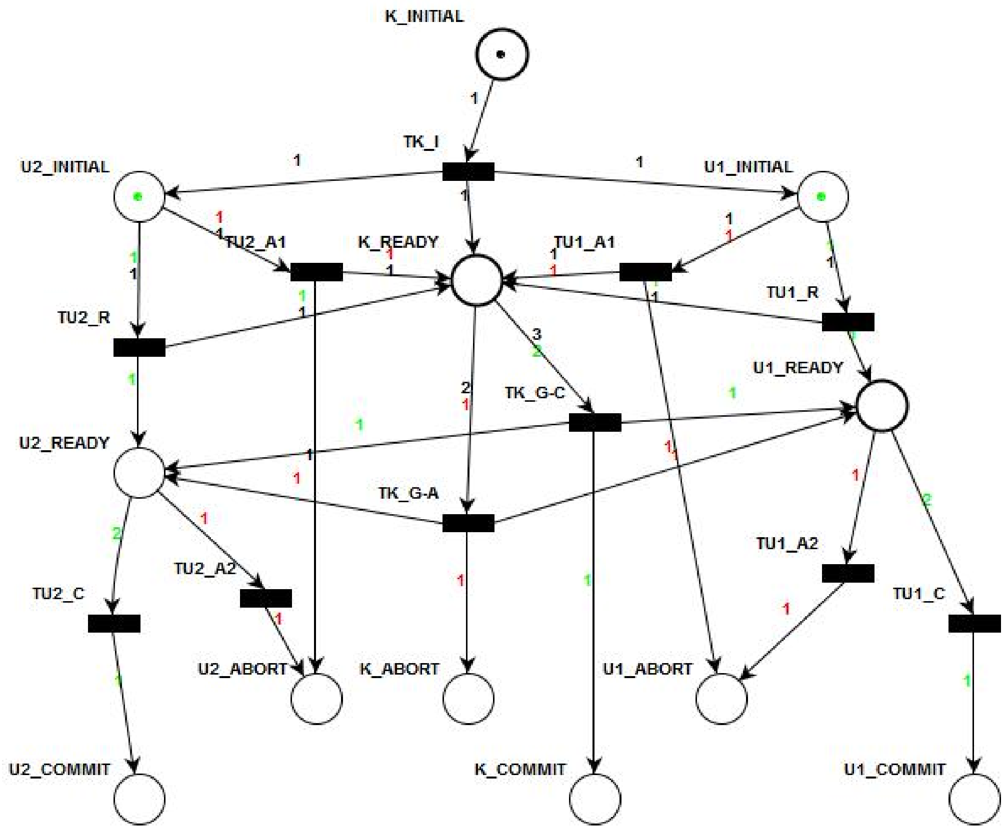

Fig. 3. Two Phase Commit protocol Petri Net model with two participant

Table 1. Coordinator places and transitions

Places		Transitions	
C_INITIAL	Initial state	TC_I	1) log entry (begin-commit) 2) message (prepare)
C_READY	Ready state	TC_G-A	1) log entry (abort) 2) message (global-abort)
C_ABORT	Global transaction abort state	TC_G-C	1) log entry (commit) 2) message (global-commit)
C_COMMIT	Global transaction commit state		

Table 2. Cohort places and transitions, upon cohort U1

Places		Transitions	
U1_INITIAL	Initial/decision state	TU1_A1	1) log entry (abort) 2) msg. to coordinator (vote-abort)
U1_READY	Ready state	TU1_R	1) log entry (ready) 2) msg. to coordinator (vote-commit)
U1_ABORT	Abort state	TU1_A2	1) log entry (abort) 2) acknowledge for coordinator
U1_COMMIT	Commit state	TU1_C	1) log entry (commit) 2) msg. to coordinator (commit)

4.1. Incidence matrixes

Table 3. Pre-incidence matrix $[D^-]$

	TC_I	TC_G-A	TC_G-C	TU1_R	TU2_R	TU1_A1	TU2_A1	TU1_A2	TU2_A2	TU1_C	TU2_C
C_INITIAL	1I										
C_READY		1A,2I	2C,3I								
C_ABORT											
C_COMMIT											
U1_INITIAL				1I,1C		1I,1A					
U1_READY								1A		2C	
U1_ABORT											
U1_COMMIT											
U2_INITIAL					1I,1C		1I,1A				
U2_READY											
U2_ABORT									1A		2C
U2_COMMIT											

Table 3 presents pre-incidence matrix which contains weights of input arcs of given transition. If arc directed from given place to given transition exists we put its weight into corresponding pre-incidence matrix cell, else we leave cell empty. If for selected arc more than one color weight is defined we put them into corresponding cell separated by coma.

Table 4 describes post-incidence matrix which contains weights of output arc of given transition. This matrix is created similarly to pre-incidence matrix, but now we are looking for arc directed from given transition to given place. Multiple colors weight on specific arc are placed into matrix cell separated by coma.

Table 4. Post-incidence matrix $[D^+]$

	TC_I	TC_G-A	TC_G-C	TU1_R	TU2_R	TU1_A1	TU2_A1	TU1_A2	TU2_A2	TU1_C	TU2_C
C_INITIAL											
C_READY	1I			1I,1C	1I,1C	1I,1A	1I,1A				
C_ABORT		1A									
C_COMMIT			1C								
U1_INITIAL	1I										
U1_READY		1A	1C	1C							
U1_ABORT						1A		1A			
U1_COMMIT										1C	
U2_INITIAL	1I										
U2_READY		1A	1C		1C						
U2_ABORT							1A		1A		
U2_COMMIT											1C

By having both pre-incidence and post-incidence matrixes we can compute incidence matrix which will allow us to perform reachability analysis.

Incidence matrix is created by substration post-incidence and pre-incidence. Matrix substration result was presented in Table 5. Created matrix defines the ways of the markings changes after each transition firing.

For example we can take first column that represents transition TC_I. The incidence matrix shows that after TC_I transition firing one INIT token (1C) will be removed from place C_INITIAL, and one INIT token (1C) will be added to places C_READY, U1_INITIAL, U2_INITIAL.

Table 5. Incidence matrix $[D] = [D^+] - [D^-]$

	TC_I	TC_G-A	TC_G-C	TU1_R	TU2_R	TU1_A1	TU2_A1	TU1_A2	TU2_A2	TU1_C	TU2_C
C_INITIAL	–1I										
C_READY	1I	–1A,–2I	–2C,–3I	1I,1C	1I,1C	1I,1A	1I,1A				
C_ABORT		1A									
C_COMMIT			1C								

Continuation of the Table 5

U1_INITIAL	1I			–1I,–1C		–1I,–1A					
U1_READY		1A	1C	1C				–1A		–2C	
U1_ABORT						1A		1A			
U1_COMMIT										1C	
U2_INITIAL	1I				–1I,–1C		–1I,–1A				
U2_READY		1A	1C		1C				–1A		
U2_ABORT							1A		1A		–2C
U2_COMMIT											1C

4.2. Reachable markings analysis

Introduction of coloured tokens into Petri Net significantly expands ability for performing simple simulations that represents scenarios that can occur during the implementation of a distributed transaction. In final markings the most significant places were bolded.

4.2.1. Optimistic case — both cohorts commits

Let's assume we want to follow the optimistic scenario in which both cohorts are ready to commit. To simulate this scenario we place COMMIT tokens at U1_INITIAL and U2_INITIAL places in initial marking. Initial and reachable marking for this case were presented in Table 6. For presentation purposes we must assume that U2 cohort will send its vote first (firing of TU2_R), but U1 cohort will be first to end its process after receiving GLOBAL-COMMIT communique (firing of TU1_C). In final marking COMMIT tokens are presented in places that correspond with successful endings of coordinator and cohorts processes.

Table 6. Reachable marking (Optimistic case — both cohorts commits)

Markings fired transitions	Places												Enabled transitions
	C_INITIAL	C_READY	C_ABORT	C_COMMIT	U1_INITIAL	U1_READY	U1_ABORT	U1_COMMIT	U2_INITIAL	U2_READY	U2_ABORT	U2_COMMIT	
M0	1I				1C				1C				TC_I
M1		1I			1C,1I				1C,1I				TU1_R,TU2_R
M2(TU2_R)		2I,1C			1C,1I					1C			TU1_R
M3(TU1_R)		3I,2C				1C				1C			TC_G-C
M4				1C		2C				2C			TU1_C,TU2_C
M5(TU1_C)				1C				1C		2C			TU2_C
M6(TU2_C)				**1C**				**1C**				**1C**	–

4.2.2. Pessimistic case 1 — one of cohorts failure

We place ABORT token (1A) into U1_INITIAL place to stimulate this place for firing of transition that represents VOTE-ABORT communique. Initial and reachable markings for this case were shown in Table 7. We assume that in voting phase cohort U2 is ready to commit and sends VOTE-COMMIT communique before U1 which detects inner error (constraint violation, resource locking) and sends VOTE-ABORT communique.

In final marking M6 tokens at C_READY and U2_READY places can reproduce partial log entries which were overwritten by more important entries about aborting distributed transaction. Two abort tokens (2A) in U1_ABORT place represent log entries about sending VOTE-ABORT communique and about GLOBAL-ABORT communique being received.

Table 7. Reachable marking (Pessimistic case 1 — one of cohorts failure)

Markings fired transitions	Places												Enabled transitions
	C_INITIAL	C_READY	C_ABORT	C_COMMIT	U1_INITIAL	U1_READY	U1_ABORT	U1_COMMIT	U2_INITIAL	U2_READY	U2_ABORT	U2_COMMIT	
M0	1I				1A				1C				TC_I
M1		1I			1A,1I				1C,1I				TU2_R, TU1_A1
M2(TU2_R)		2I,1C			1A,1I					1C			TU1_A1
M3(TU1_A1)		3I,1C,1A					1A			1C			TC_G-A
M4		1I,1C	1A			1A	1A			1A, 1C			TU1_A2, TU2_A2
M5(TU2_A2)		1I,1C	1A			1A	1A			1C	1A		TU1_A2
M6(TU1_A2)		1I,1C	**1A**				**2A**			1C	**1A**		–

4.2.3. Pessimistic case 2 — coordinator failure

Table 8. Reachable marking (Pessimistic case 2 — coordinator failure)

Markings fired transitions	Places												Enabled transitions
	C_INITIAL	C_READY	C_ABORT	C_COMMIT	U1_INITIAL	U1_READY	U1_ABORT	U1_COMMIT	U2_INITIAL	U2_READY	U2_ABORT	U2_COMMIT	
M0	1I	1A			1C				1C				TC_G-A, TU1_I, TU2_I
M1(TC_G-A)			1A		1C,1I	1A			1C,1I	1A			TU1_A2, TU2_A2
M2(TU1_A2)			1A		1C,1I		1A		1C,1I	1A			TU2_A1
M3(TU2_A2)			**1A**		1C,1I		**1A**		1C,1I		**1A**		

During the analysis of commit protocol we must consider important pessimistic scenario in which the coordinator failure occurs. Example situation was shown in Table 8. In this case the cohorts were ready to commit and coordinator failure happened after sending initial communique. For clear presentation Table 8 is not showing firing of transitions TU1_I and TU2_I. After coordinator detects its failure transition TC_G-A is being fired resulting coordinator state change into C_ABORT. It is also causing appearance of ABORT tokens in places TU1_R and TU2_R which results immediate firing of transitions responsible of aborting cohort processes (TU1_A, TU2_A).

Resume

In this work Coloured Petri Net model of 2PC protocol with two participants was presented. With usage of reachability analysis the selected scenarios of distributed transactions were shown. In future research works we plan to run a test distributed database environment. The environment will be created with the usage of database engines that support X/Open XA standard which uses 2PC at its core. We are planning an implementation of distributed transaction coordinator application in JAVA language. The Coloured Petri Net model presented here will be compiled with the experience we will gain during the test environment creation and the coordinator implementation. Future work will help us in developing new conclusions about mathematical theory of Petri Nets in database science.

1. *Iwaniak M.* Distributed Transactions Modeling with the Use of Petri Nets / M. Iwaniak, W. Khadzhynov // Data Rec., Storage & Processing. — 2012. — Vol. 14, N 3. — P. 81–91.

2. *M. Tamer Özsu.* Principles of Distributed Database Systems. — [III ed.] / M. Tamer Özsu, Patrick Valduriez. — Springer, 2011.

3. *Banaszak Z.* Procesy Współbieżne: Modele Efektywności Funkcjonowania / Z. Banaszak, P. Majdzik, R. Wójcik. — Koszalin: Politechnika Koszalińska, 2011.

4. *Bidyut Biman Sarkar.* Transaction Management for Distributed Database Using Petri Nets / Bidyut Biman Sarkar, Nabendu Chaki // International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM). — 2010. — Vol. 2. — P. 069–076. — ISSN: 2150-7988.

5. *Bidyut Biman Sarkar.* Virtual Data Warehouse Modeling Using Petri Nets for Distributed Decision Making / Bidyut Biman Sarkar, Nabendu Chaki. — doi:10.4156/jcit.vol5.issue5.1.

Received 05.09.2013