IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 1, January 2017

Analysis and Simulation of a Cookie Vending Machine Model based on Stochastic Petri Nets

Sofre Garevski[1], Emilija Spasova Kamceva[2]

Bachelor's degree, Faculty of Informatics, FON University, Skopje, Republic of Macedonia [1]

Ph. D, Faculty of Informatics, FON University, Skopje, Republic of Macedonia [2]

Abstract: Petri nets are a well-known formalism for the description of concurrency and synchronization inherent in modern distributed systems. The graphical representation of Place-Transition net is called model. In order to understand the properties of the model and to be sure that it behaves as it should, the model can be analysed and simulated. In this paper we are going to show simple model for the Cookie Vending Machine, designed with stochastic petri nets. Then we are going to analyse and simulate the model using the Petri Net Toolbox in Matlab. At the end the collected results from analyses will be discussed.

Keywords: Analysis, Simulation, Cookie Vending Machine, Petri Net, Stochastic Petri Net, Petri Net Toolbox.

I. INTRODUCTION

The term Petri Net origins from Carl Adam Petri's dissertation [1] submitted in 1962 to the faculty of Mathematics and Physics at the Technical University of Darmstadt, West Germany. Development and applications of Petri Nets date from 1970, when the Computation Structure Group at MIT was most active in conducting Petri-net related research, and produced many reports and thesis on Petri Nets. In that time European researchers were very active in organizing workshops and publishing conference proceedings on Petri nets. Later in 1985 another series of international workshops initiated, which target timed and stochastic nets and their applications to performance evaluation [2].

Petri Nets are graphical and mathematical promising tool, for describing and studying systems which can have properties like concurrency, can be asynchronous, distributed, nondeterministic and/or stochastic. Also as a graphical tool can be used similar to flow charts, block diagrams, and networks. Because of this, they have been proposed for a very wide variety of applications and any area that can be described graphically and that needs representation of parallel or concurrent activities. Some application areas are: performance evaluation, communication protocols, distributed-software systems, distributed-database systems, concurrent and parallel programs, flexible manufacturing/industrial control systems, discrete event systems and etc. [2].

The use of computer-aided tolls is needed for practical application of Petri nets. Most of the research groups use their own tools to analyse and simulate various kinds of Petri nets. References where Petri nets are applied in practice [3]-[5], and commercial and open source tools currently available [6] are given in [7]. Common technics for analysing various types of Petri nets [8] are given in [9].

The rest of this paper is organized as follows: in section A is given repetition of Petri nets, its formal definition and informally the transition enabling and firing rule. Also are given formal definition for four types of analysis: marking graph (covering tree), traps and cotraps, place invariants and transition invariants.

These analyses are then applied on the Cookie Vending Machine model in section B. There will be used the Petri Net Toolbox tool to analyse and simulate the model defined with stochastic Petri nets. Finally, section C contains discussion about the collected results.

II. OVERVIEW

The graphical representation of a Place-Transition net com-prises the following components:

- Places drawn by circles. Places model conditions or objects, e.g., a program variable.
- Tokens drawn by black dots. Tokens represent the specific value of the condition or object, e.g., the value of a program variable.
- Transitions drawn by rectangles. Transitions model activities which change the values of conditions and objects.
- Arcs, specifying the interconnection of places and transitions thus indicating which objects are changed by a certain activity.

A Petri net is a particular kind of directed graph, together with an initial state called the initial marking M_0. The underlying graph N of a Petri net is a directed, weighted, bipartite graph consisting of places and transitions, where arcs are either from a place to transition or from a transition to a place. This is presented on Figure 1.

DOI 10.17148/IJARCCE.2017.6112

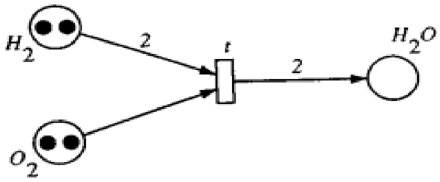


Fig. 1 Example petri net

Ordinary Petri net (definition) - An Ordinary Petri net is one where all arcs are unity-weighted (and hence unlabelled), mathematically represented by a four-tuple N = (P, T, W^-, W^+) where $\forall p \in P$, $\forall t \in T$, $W^-(p,t) \le 1$ and $W^+(p,t) \le 1$.

Marked Petri net (definition) - Marked Petri net is a five tuple N = (P,T, W^-, W^+,M_0) where:

- P = {p_1, … p_n} is a finite and non-empty set of places,
- T = {t_1, … t_m} is a finite and non-empty set of transitions,
- P ∩ T = ∅,
- W^-, W^+: P × T → N_0 are the backward and forward incidence functions, respectively
- M_0: P → N_0 is the initial marking.

Stochastic Petri net(definition) - Stochastic Petri net[13]is a form of Petri net where the transitions fire after a probabilistic delay determined by a random variable.

The net on Figure 1 is Marked and not ordinary Petri net and can be described as: P = {H_2, O_2, H_2O}, T = {t}, M_0 = {2, 2, 0}, W^- = [(H_2, t), (O_2, t)] = [2, 1], W^+ = [(t, H_2O)] = [2]. W^- and W^+ represent the arcs from places to transitions and from transitions to places accordingly.

Execution of a petri net consists of tokens that travel from one or multiple states through transition to some other state or multiple states. This is called firing of transition and as result the marking of the places in the petri net changes. A state or marking of Petri nets is changed according to the following transition firing rule:

- A transition t is said to be enabled if each input place p of t is marked with at least w(p, t) tokens, where w(p, t) is the weight of the arc from p to t
- An enabled transition may or may not fire (depending on whether or not the event actually takes place).
- A firing of an enabled transition t removes w(t, p) to-kens to each output place p of t, where w(t, p) is the weight of the arc from t to p.

Figure1 and Figure2 are illustrating the transition rule, using the well-known chemical reaction: $2H_2 + O_2 \rightarrow H_2O$. Two tokens in each input place in Figure 1 show that two units of H_2 and O_2 are available, and the transition t is enabled. After firing t, the marking will change to the one shown in Figure 2, where the transition t is no longer enabled [2].

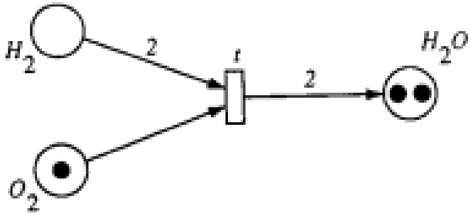


Fig. 2 An illustration of transition firing rule. The marking after firing t, where t is disabled

In Figure 1 and Figure 2 the possible markings can be M_0 also called initial marking, and after execution of t the new marking M_1 = {0 ,1, 2}. Real-world systems described with Petri nets can have much more states and transitions. Often it is difficult to predict if some marking can be reached, or some transition can fire. This is connected with the term state properties and can be determined with applying some analysis techniques which will be discussed in the next section.

A. Analysis techniques

There are some very common properties that are often asked about in a system net N: does N terminate? Can N reach a marking that no longer enables any transitions? Can each transition always become enabled again? From each reachable marking, is it possible to reach the initial marking again? Some of these questions can be answered with the marking graph(covering tree).

Marking graph: For a system net N and an initial marking M_0, a marking M of N is reachable if there exists a sequence of steps σ with M_n= M. The reachable markings and steps of a system net N can be compiled in marking graph G of N [9]. Most of the properties of the system net N can be defined and characterize them as properties of the marking graph G.

Properties of a Petri net that are dependent on the initial marking are referred to as behavioral properties. Those that are independent of the initial marking and dependent only on the structure of the Petri net are called structural properties [10, 11]. Behavioural properties are:

- Reachability (definition) - The marking M_n is said to be reachable from M_0 if there exists firing sequence σ that will yield M_n .

- Boundedness (definition) - A Petri net is **k-bounded** with respect to an initial marking if the number of tokens in any of its places never exceeds k for any marking in the reachability set $R(M_0)$, i.e. $M(p) \le k$, $\forall p \in P$ and $\forall M \in R(M_0)$, where $M(p)$ is the number of tokens in place p in marking M.

- Safe (definition) - A Petri net is **safe** if it is k-bounded and k=1

- Liveness (definition) - A Petri net is said to be **live** if for all transitions there is a way to fire transition in any marking M' reachable from the initial marking M_0 .

DOI 10.17148/IJARCCE.2017.6112

- Deadlock (definition) - A marking $M^{'}$ reachable from the initial marking M_0 is a **deadlock** if none of the transitions of the Petri net is enable in $M^{'}$.
- Reversibility (definition) - A Petri net N with initial state M_0 is said to be **reversible** if for each marking $M \in R(M_0)$, M_0 is reachable from M.
- Coverability (definition) - A marking M in a Petri net N with initial state M_0 is said to be **coverable** if, there exists a marking $M^{'} \in R(M_0)$ such that $M^{'}(p) \geq M(p)$ for all $p \in P$.
- Persistence (definition) - A Petri net is said to be **persistent** if, for any two enabled transitions, occurrence of one transition will not disable another.

Structural properties of Petri nets include:
- Structurally live (definition) - A Petri net is said to be **structurally live** if there exists an initial marking M_0 such that net is live. A Petri net which is live is also structurally live, but the reciprocal is false.
- Structurally bounded (definition) - A Petri net is said to be **structurally bounded** if it is bounded for any initial marking M_0 . Structural boundedness requires that the net remains bounded for all possible initial marking.
- Conservativiness (definition) - A Petri net is **conservative**, if all transitions fire token-preservingly, i.e. all transitions add exactly as many tokens to their output places as they subtract from their input places.
- Repetitivness (definition) - A Petri net is said to be **repetitive** if there exists an initial marking M_0 and a firable sequence σ in which each transition appears an unlimited number of times.

Traps and Cotraps: Trap of an elementary system net N is a subset Q of the places of N such that for each transition t of N the following holds:

If there exists place p ∈ Q such that p ∈ ${}^\bullet t$, then there also exists a place q ∈ Q such that q ∈ $t^\bullet$. (1)

A trap Q of an elementary system net N is called marked ina marking M if there exists at least one place $q \in Q$ such that $M(q) \geq 1$. If $Q = \{q_1,...,q_r\}$ is a trap marked in M, then the following holds:

$$M(q_1)+...+M(q_r) \geq 1 \, (2)$$

For each step $M \xrightarrow{t} M'$ of N there exist only two possibilities: either there exists a place $p \in Q$ such that $p \in {}^\bullet t$, in which case, according to (1), there also exists a place $q \in Q$ such that $q \in t^\bullet$. Then $M'(q) \geq 1$ and thus

$$M'(q_1)+...+M'(q_r) \geq 1 \ (3)$$

Or there exists no such place p [9].

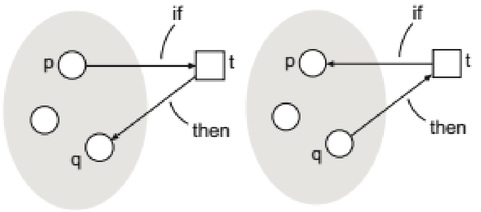


Fig. 4 Trap {p,q…} and cotrap {p,q…} accordingly

As a counterpart to a trap, a cotrap of an elementary system net N is a subset Q of the places of N such that for each transition t of N the following holds:
If there exists a place p ∈ Q such that p ∈ $t^\bullet$,then there also exists a place q ∈ Q such that q ∈ ${}^\bullet t$. (4)
Let $Q = \{q_1,...,q_n\}$ be a cotrap that is unmarked in a marking M, that is

$$M(q_1)+...+M(q_n) = 0 \, (5)$$

Let $M \xrightarrow{t} M'$ be a step. Then for each place $p \in {}^\bullet t : M(p) \geq 1$. This means, because of Eq. (5), that there cannot exist a place $q \in Q$ such that $Q \in {}^\bullet t$. Because of (4), there also cannot exist a place $p \in Q$ such that $p \in t^\bullet$ [9]. Then it follows from Eq. (5) that

$$M'(q_1)+...+M'(q_n) = 0 \, (6)$$

Place and Transition invariants: The markings in elementary Petri net can be written as column vector

$$\underline{M} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix}$$

Where a$_i$is the number of tokens in the I th place. Similar transition t can be also written as vector

$$\underline{t} = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} \text{ where } \underline{z_i} = def \begin{cases} -1, & if \ \ p_i \in {}^\bullet t \, and \ p_i \notin t^\bullet, \\ +1, & if \ \ p_i \in t^\bullet \ and \ p_i \notin {}^\bullet t, \\ 0, & otherwise \end{cases}$$

for $i = 1,...,k$. Step $M \xrightarrow{t} M'$can be expressed as $\underline{t}$. This is called vector representation of steps [9]. For each transition t_i in Petri net N can be created vector $\underline{t_i}$, which together form the matrix $\underline{N}$ (incidence matrix) and can be defined as

$$\underline{N} = def\,(t_1,...,t_l) = \begin{pmatrix} z_{11} & \cdots & z_{l1} \\ \vdots & & \vdots \\ z_{1k} & \cdots & z_{lk} \end{pmatrix}$$

Also $\underline{N}$ can be calculated from the backward and forward incidence matrices:

$$\underline{N} = W^{+} - W^{-} \ \ (7)$$

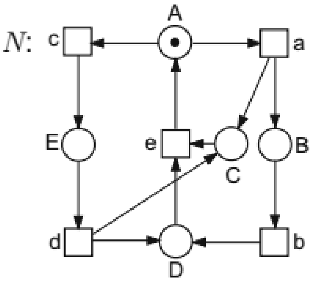
Valid equations can be derived from solutions of the system of equations

$$\underline{x} \cdot \underline{N} = \vec{0} \quad (8)$$

where $\vec{0} = (0,...,0)$ is an l-dimensional row vector. A vector $\underline{n} = (n_1,...,n_k)$ solves (8) if and only if $\underline{n} \cdot \underline{t} = 0$ for each transition t of N . A solution $\underline{n}$ of (8) is called a place invariant of N .Similar if we calculate incidence matrix $\underline{N}$ as

$$\underline{N} = W^- - W^+ \quad (9)$$

then every solution $\underline{n}$ of (8) is called transition invariantof N .



$W^+ - W^-$	a	b	c	d	e
A	-1	0	-1	0	1
B	1	-1	0	0	0
C	1	0	0	1	-1
D	0	1	0	1	-1
E	0	0	1	-1	0

Fig. 5 Example Petri net and its incidence matrix

On Figure 5 is displayed example Petri net with its incidence matrix $\underline{N} = W^+ - W^-$. From here and (8) can be constructed the place invariant equations

$$\begin{cases} -x_1 + x_2 + x_3 = 0 \\ -x_2 + x_4 = 0 \\ -x_1 + x_5 = 0 \\ x_3 + x_4 - x_5 = 0 \\ x_1 - x_3 - x_4 = 0 \end{cases} \equiv \begin{cases} x_2 = x_4 \\ x_1 = x_5 \\ x_1 - x_2 - x_3 = 0 \end{cases}$$

and for $x_1 = 1$ will get place invariants i_1 and i_2

$W^+ - W^-$	i_1	i_2
A	1	1
B	1	0
C	0	1
D	1	0
E	1	1

Now that we know the techniques for analysing Petri nets we are ready to analyse the “Cookie Vending Machine Model”.

B. The Cookie Vending Machine Model

The cookie vending machine consists of a coin slot and a compartment into which the packets of cookies are dropped. It has the ability to return the inserted coin and refuel its storage if empty. This functionality is modeled with Stochastic Petri net in Figure 6.

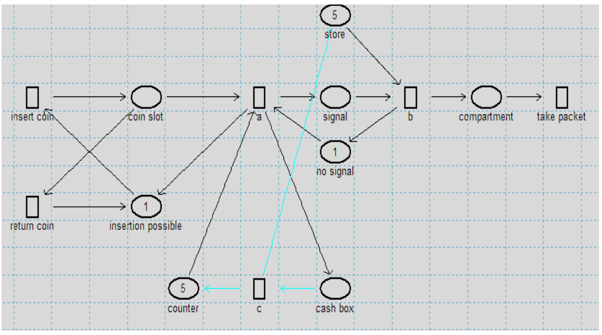


Figure 6. The cookie vending machine model

At the beginning the state “counter” and the state “storage” contains by five tokens and the insertion of a coin is possible. All of the transitions happen with exponential probability of 0.3. The arcs from state “cash box” to transition “c” and from transition “c” to states “counter” and “storage” have weight of 5.

All other arcs have weight of 1. After insertion of a coin the state “coin slot” gets one token, the state “insertion possible” loses one token and the transitions “a” and “return coin” becomes concurrent. If any of these two states happens the insertionof coin will be enabled again. After the execution of transition “a”, the state ”counter” and “no signal” loses by one token and the states ”cash box” and “signal” are getting by one token. This enables transition “b” which returns by one token to the states “no signal” and “compartment” and removes one token from the state “storage”. After five executions of transition “a”, the transition “c” becomes enabled and the states “counter” and “storage” will get by five tokens.The current implementation, allows execution of the transition “insert coin” while transition “take packet” has not been fired. This is not case in the real world system, but the example is adjusted for analysis and simulation purposes.

C. Analysis

The model on Fig. 6 is Markednot ordinary Petri net (P, T, W$^-$, W$^+$,M$_0$) which comprises of:

P: {coin slot, insertion possible, counter, storage, signal, no signal, cash box, compartment},

T: {insert coin, return coin, a, c, b, take packet},

W$^-$: {(coin slot, return coin), (coin slot, a), (insertion possible, insert coin), (counter, a), (storage, b), (signal, b), (no signal, a), (cash box, c), (compartment, take packet)} = {(-1), (-1), (-1), (-1), (-1), (-1), (-1), (-5), (-1)},

W$^+$: {(insert coin, coin slot), (return coin, insertion possible), (a, insertion possible), (a, cash box), (a, signal), (c, counter), (c, storage), (b, no signal), (b, compartment)} = {(1), (1), (1), (1), (1), (5), (5), (1), (1)},

DOI 10.17148/IJARCCE.2017.6112

M$_0$: (0, 1, 5, 5, 0, 1, 0, 0).
In order to analyze and simulate the model we have used the Petri Net Toolbox tool in Matlab. Most of the behavioral and structural properties can be determined from the coverability tree (Marking graph) which is presented in Figure 7. The Structural properties of the net are:

- The net is **unbounded** (unbounded place "compartment")
- The net is **partially conservative**. Places that do not belong to any P-invariant support: compartment
- The net is **repetitive**
- The net is **consistent**

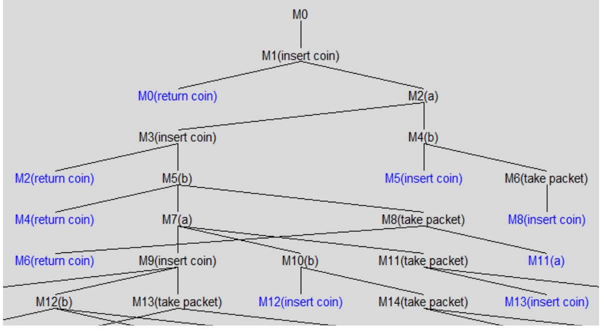


Fig. 7. Covering tree

All of the places are bounded except the "compartment". The following equations hold for the bounded places:

- coin slot + insertion possible = 1
- signal + no signal = 1
- counter + cash box = 5
- storage + no signal + cash box = 6

From (9) can be calculated the incidence matrix $\underline{N}$ (Fig. 8). Using (8) can be constructed the place invariant and transition invariants equations (Fig.9.a and 9.b accordingly).

W	insert coin	return coin	a	c	b	take packet
coin slot	1	-1	-1	0	0	0
insertion possible	-1	1	1	0	0	0
counter	0	0	-1	5	0	0
storage	0	0	0	5	-1	0
signal	0	0	1	0	-1	0
no signal	0	0	-1	0	1	0
cash box	0	0	1	-5	0	0
compartment	0	0	0	0	1	-1

Fig.8 Incidence matrix for the cookie vending machine

$$a)\begin{cases} x_1 - x_2 = 0 \\ -x_1 + x_2 = 0 \\ -x_1 + x_2 - x_3 + x_5 - x_6 + x_7 = 0 \\ 5x_3 + 5x_4 - 5x_7 = 0 \\ -x_4 - x_5 + x_6 + x_8 = 0 \\ -x_8 = 0 \end{cases} \equiv \begin{cases} x_1 = x_2 \\ x_3 + x_4 = x_7 \\ x_4 + x_5 = x_6 \\ x_8 = 0 \end{cases}$$

$$b)\begin{cases} x_1 - x_2 - x_3 = 0 \\ -x_1 + x_2 + x_3 = 0 \\ -x_3 + 5x_4 = 0 \\ 5x_4 - x_5 = 0 \\ x_3 - x_5 = 0 \\ -x_3 + x_5 = 0 \\ x_3 - 5x_4 = 0 \\ x_5 - x_6 = 0 \end{cases} \equiv \{x_3 = x_5 = x_6 = 5x_4 = x_1 - x_2$$

Fig.9.a Place invariant equations.9.b Transition invariant equations

At most four place invariants are linearly independent and those are shown in Figure 10.

P – invariants	i_1	i_2	i_3	i_4
coin slot	1	0	0	0
insertion possible	1	0	0	0
counter	0	1	0	0
storage	0	0	0	1
signal	0	0	1	0
no signal	0	0	1	1
cash box	0	1	0	1
compartment	0	0	0	0

Fig. 10 Place invariants

From the place invariants table, we prove that every state that belongs to some invariant is bounded, and this is the case for all of the states except for the state "compartment". At most two transition invariants are linearly independent and those are shown in Figure 11.

T – invariants	i_1	i_2
insert coin	1	5
return coin	1	0
a	0	5
c	0	1
b	0	5
take packet	0	5

Fig. 11 Transition invariants

From the transition invariants table can be easy recognized the weight of the transitions. For example, if one execution of the transition "c" happens, this means that previously transitions "insert coin", "a", "b" and "take packet" have been executed five times.

D. Simulation
Our stochastic model can be viewed as a queue with exponentialarrival A(t) distribution, exponential service distribution B(x) and one server (M/M/1 queue). Scheduling strategy at the server is FCFS (First Come First Served) and are used 1000 samples (customers). Important outputs of the simulation are:

-Arrival rate (λ) defined as $\lambda = \frac{1}{\int_0^\infty t dA(t)} = \frac{1}{\int_0^\infty t de^t}$

- Service rate (μ) defined as $\mu = \frac{1}{\int_0^\infty x dB(x)} = \frac{1}{\int_0^\infty x de^x}$
- Service sum - total number of firings during the simulation
- Service distance - the mean time between two successive firings
- Utilization - the fraction of time when server is busy

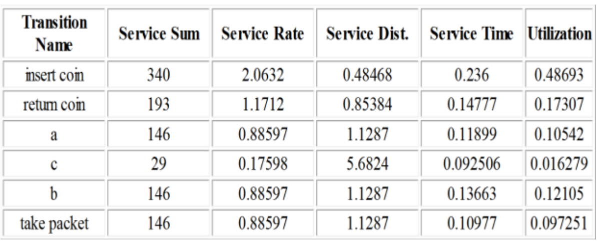
For the places important outputs are:

- Arrival sum - the total number of arrived tokens
- Throughput sum - the total number of departed tokens
- Arrival distance - the mean time between two successive instants when tokens arrive to the place
- Throughput distance - the mean time between two successive instants when tokens depart from the place
- Waiting time - the mean time a token spends in a place
- Queue length - the average number of tokens weighted by time

In order to see the execution of the transitions we simulated the model in three cases:

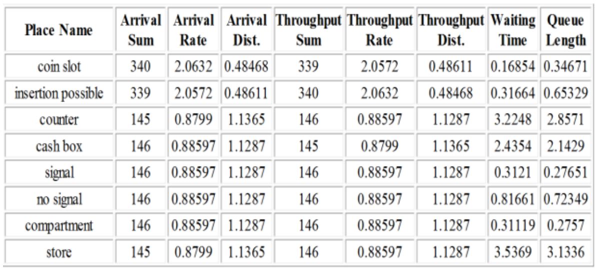
I. All of the transitions with same probability of 0.3
II. Probability “1” for all transitions
III. Probability “1” for the transitions “insert coin” and “take packet” all others with probability 0.3

Transition indices for the simulation (I), (II) and (III) are shown on Figure 12-17 accordingly.



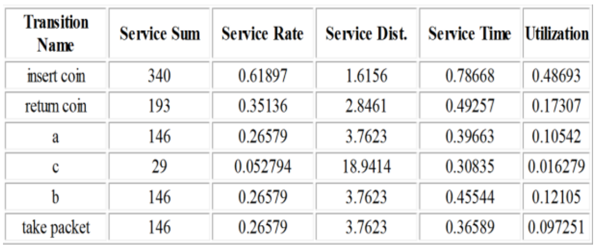
Transition Name	Service Sum	Service Rate	Service Dist.	Service Time	Utilization
insert coin	340	2.0632	0.48468	0.236	0.48693
return coin	193	1.1712	0.85384	0.14777	0.17307
a	146	0.88597	1.1287	0.11899	0.10542
c	29	0.17598	5.6824	0.092506	0.016279
b	146	0.88597	1.1287	0.13663	0.12105
take packet	146	0.88597	1.1287	0.10977	0.097251

Figure 12. Transition indices I



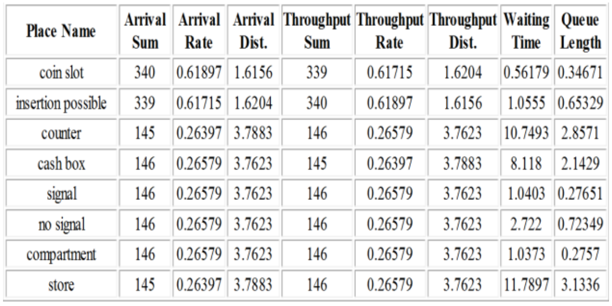
Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
coin slot	340	2.0632	0.48468	339	2.0572	0.48611	0.16854	0.34671
insertion possible	339	2.0572	0.48611	340	2.0632	0.48468	0.31664	0.65329
counter	145	0.8799	1.1365	146	0.88597	1.1287	3.2248	2.8571
cash box	146	0.88597	1.1287	145	0.8799	1.1365	2.4354	2.1429
signal	146	0.88597	1.1287	146	0.88597	1.1287	0.3121	0.27651
no signal	146	0.88597	1.1287	146	0.88597	1.1287	0.81661	0.72349
compartment	146	0.88597	1.1287	146	0.88597	1.1287	0.31119	0.2757
store	145	0.8799	1.1365	146	0.88597	1.1287	3.5369	3.1336

Figure 13. Place indices I



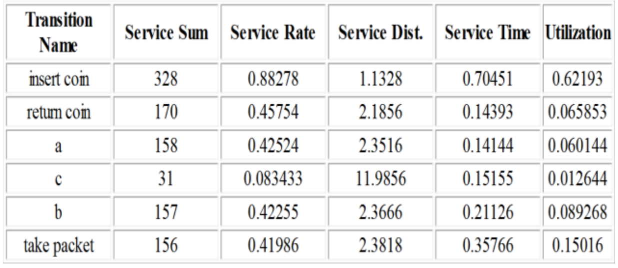
Transition Name	Service Sum	Service Rate	Service Dist.	Service Time	Utilization
insert coin	340	0.61897	1.6156	0.78668	0.48693
return coin	193	0.35136	2.8461	0.49257	0.17307
a	146	0.26579	3.7623	0.39663	0.10542
c	29	0.052794	18.9414	0.30835	0.016279
b	146	0.26579	3.7623	0.45544	0.12105
take packet	146	0.26579	3.7623	0.36589	0.097251

Figure 14. Transition indices II



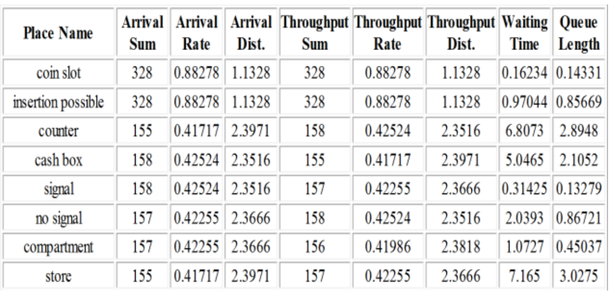
Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
coin slot	340	0.61897	1.6156	339	0.61715	1.6204	0.56179	0.34671
insertion possible	339	0.61715	1.6204	340	0.61897	1.6156	1.0555	0.65329
counter	145	0.26397	3.7883	146	0.26579	3.7623	10.7493	2.8571
cash box	146	0.26579	3.7623	145	0.26397	3.7883	8.118	2.1429
signal	146	0.26579	3.7623	146	0.26579	3.7623	1.0403	0.27651
no signal	146	0.26579	3.7623	146	0.26579	3.7623	2.722	0.72349
compartment	146	0.26579	3.7623	146	0.26579	3.7623	1.0373	0.2757
store	145	0.26397	3.7883	146	0.26579	3.7623	11.7897	3.1336

Figure 15. Place indices II



Transition Name	Service Sum	Service Rate	Service Dist.	Service Time	Utilization
insert coin	328	0.88278	1.1328	0.70451	0.62193
return coin	170	0.45754	2.1856	0.14393	0.065853
a	158	0.42524	2.3516	0.14144	0.060144
c	31	0.083433	11.9856	0.15155	0.012644
b	157	0.42255	2.3666	0.21126	0.089268
take packet	156	0.41986	2.3818	0.35766	0.15016

Figure 16. Transition indices III



Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
coin slot	328	0.88278	1.1328	328	0.88278	1.1328	0.16234	0.14331
insertion possible	328	0.88278	1.1328	328	0.88278	1.1328	0.97044	0.85669
counter	155	0.41717	2.3971	158	0.42524	2.3516	6.8073	2.8948
cash box	158	0.42524	2.3516	155	0.41717	2.3971	5.0465	2.1052
signal	158	0.42524	2.3516	157	0.42255	2.3666	0.31425	0.13279
no signal	157	0.42255	2.3666	158	0.42524	2.3516	2.0393	0.86721
compartment	157	0.42255	2.3666	156	0.41986	2.3818	1.0727	0.45037
store	155	0.41717	2.3971	157	0.42255	2.3666	7.165	3.0275

Figure 17. Place indices III

From transition and place tables with indices can be identified that values for some states or transitions have similar value. From 1000 samples, more than 1/3 are “insert coin”. The indices for transitions “a”, “b” and “take packet” have similar values. The same goes for the places “coin slot” and “insertion possible” as a group, and places “counter”, “cashbox”, “signal”, “no signal”, “compartment” and “store” as a group.

III.CONCLUSION

Petri nets are well known formalism for describing processes. The properties and behaviour of them can be analysed and synchronized using computer tools. We applied the Marking graph, Traps and Cotraps and Places and Transition invariants techniques on the Cookie Vending Machine model designed with stochastic Petri nets.

To confirm this and to analyse the model we used the Petri net Toolbox. The tool was not able to return some properties for the net because it was ordinary. At the end we simulated the model for three different scenarios with different probabilities which resulted with similar output values.

DOI 10.17148/IJARCCE.2017.6112

REFERENCES

[1] Petri, Carl Adam. "Kommunikation mit automaten." (1962).
[2] Murata, Tadao. "Petri nets: Properties, analysis and applications." Proceedings of the IEEE 77.4 (1989): 541-580.
[3] http://www.informatik.uni-hamburg.de/TGI/PetriNets/applications/
[4] http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/java/
[5] http://www.informatik.uni-hamburg.de/TGI/PetriNets/bibliographies/
[6] http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/
[7] https://www.informatik.uni-hamburg.de/TGI/PetriNets/
[8] http://www.informatik.uni-hamburg.de/TGI/PetriNets/classification/
[9] Reisig, Wolfgang. Understanding Petri Nets. Vol. 4. Springer, Berlin, 2013.
[10] http://brahms.emu.edu.tr/rza/isys501week2.pdf
[11] http://brahms.emu.edu.tr/rza/isys501week3.pdf
[12] Li, ZhiWu, and MengChu Zhou. Deadlock resolution in automated manufacturing systems: a novel Petri net approach. Springer Science & Business Media, 2009.
[13] Marsan, M. Ajmone. "Stochastic Petri nets: an elementary introduction." European Workshop on Applications and Theory in Petri Nets. Springer Berlin Heidelberg, 1988.

DOI 10.17148/IJARCCE.2017.6112