Journal of Artificial Intelligence in Electrical Engineering, Vol. 2, No. 6, August 2013

Modeling a Bank ATM with Two Directions Places Timed Petri Net (TPN)

Nasrin Fatholahzadeh
Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
Email: n-fatholahzadeh@iau-ahar.ac.ir

ABSTRACT
A Bank ATM is including controller, card authorization system and a teller unit .This paper explains how this parts connects together. In this paper will be used of a new mode place in Petri nets. More systems usually have a complex constructs. ATM will be simulated use of new mode place at this study. The Main part of this model is used of T.S.Staines model [1]. We discuss how a simple model of T.S.Staines model be used for new model in this study. At this work will be added money teller unit that was discussed in T.S.Staines model. Each main components of the system are identified and built using Petri Nets. The final Model is live and exhibits repetitive, consistent behavior. The work presented here is continue of last work and can be further developed.

1. INTRODUCTION

A bank ATM built of computerized machinery composed of difference classes of objects: Display Unit, Communication component, Printer, Keypad and etc. for good working must be all these components synchronize with another component. Each of those simple transactions in reality is composed of several steps. Some transaction such as communication and authorization are controlled remotely from the ATM. Various diagrams can be used to model this system like the Activity Diagram, Sequence Diagram, Communication diagram or State Diagram of the UML 2 [2]. To properly understand the system being explored the notion of synchronization of events and objects are of fundamental importance. For Real Time behavior complex mechanisms, redundancy and time must be considered. There is also a problem with diagram consistency if too many diagrams are used. The bank ATM can be considered to be a soft Real - Time system which requires some form of performance analysis. Even though there are numerous methods that can be used for analysis, often it is better to keep to one method rather than end up with a collection of methods .There are formal techniques like: Real Time Logics, Duration Calculus, Process Algebras and Formal languages [3]. Usually there is no visual depiction of the system. It is possible to have a formally correct system exhibiting unwanted behavior if this is incorrectly modeled. For a bank ATM, there are many possible events and states. Understanding this system is highly dependent on dynamic modeling. A person's natural thought process must naturally focus on the interaction between the system's structure and events not on keeping models consistent. This can be achieved through the use of Petri Nets. Some of the most common classes of Petri

Nets are Timed Petri Nets, Stochastic Petri Nets, Generalized Stochastic Petri Nets and colored Petri Nets. Timed Petri Nets have been used successfully to model network ATM switching. Special firing times and timing issues have been used in these cases. Transitions can have deterministic or exponential firing times [4].

High level Petri Nets have been used to describe intelligent network switching. A Petri Net can perform some form of choice using defined conditions. Timed Petri Nets, Stochastic Petri Nets and Generalized Stochastic Petri Nets are special classes that can be used for the design of real time systems and embedded systems [5]. Petri Nets can be used in a three-view model: path-view, map-view and resource view [6]. This is useful for performance engineering of concurrent software. These models like TCPN (Timing Constraint Petri Nets) can be used for schedulability analysis of real time system specifications. Most of the work done with Petri Nets does not usually take a complete system into consideration from a highly detailed point of view. Also it is not usual for these models to cater for redundancy. This paper shows how many other details and mechanisms can be included in the Petri Nets being used. These mechanisms were built using the Petri Net theory. These mechanisms or constructs cannot be easily replicated using other methods.

2. RELATED WORK

This model for bank ATM has 4 main parts. Refer to Fig. 1. These are:

I) The Bank ATM machine

ii) An ATM Controller

iii) Credit card authorization server

iv) Money teller unit

The ATM machine, the ATM Controller and credit card authorization with money teller unit work together synchronic. We describe briefly the work carried out:

I) Events and states were identified by 4 main parts. E.g. for the ATM possible events are: Customer Arrives, Insert Card, Insert Pin, Validate Pin , peek money , save money and others.

ii) These events were used to build four Timed Petri Nets, one of them for ATM Machine, second for the ATM Controller, other for the Card Authorization System and latest for teller unit.

iii) Many additional mechanisms were added to the basic Petri Nets to make their behavior more realistic.

iv) The Petri Nets were tested in isolation for all parts for any kind of errors. v) At the end four Petri Nets model for all parts were connected to produce the final system model.

vi) The full system Petri Net model was tested and simulated to derive results used for performance analysis. Three first parts are used of T.S.Staines model. For use of them need to simple model of them instead full model. This simple model must be of work like to full model.

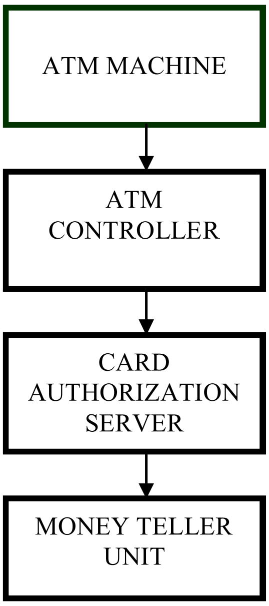

Fig. 1. The main parts for a bank ATM

2.1 T.S.STAINES'S SIMPLE MODEL

Fig. 2. Shows this simple model. Processing result for simple model almost equal with processing result full model. The final model resulted in Timed Petri Net. This was used to obtain performance results. Two types of transitions were used:

I) time transitions

ii) Immediate transitions

The final results appear for performance results in Table 1.

Table 1: Processing results for 8 runs of the T.S.Staines simple model

Run No	Total cards Taken invalid PIN	Timeouts Inserting PIN	Timeout Transaction selection	Total ATM Controller Timeout
1	326	349	97	11
2	330	350	94	11
3	335	348	95	10
4	330	347	95	10
5	330	351	93	12
6	333	347	97	10
7	331	355	93	11
8	334	349	95	11

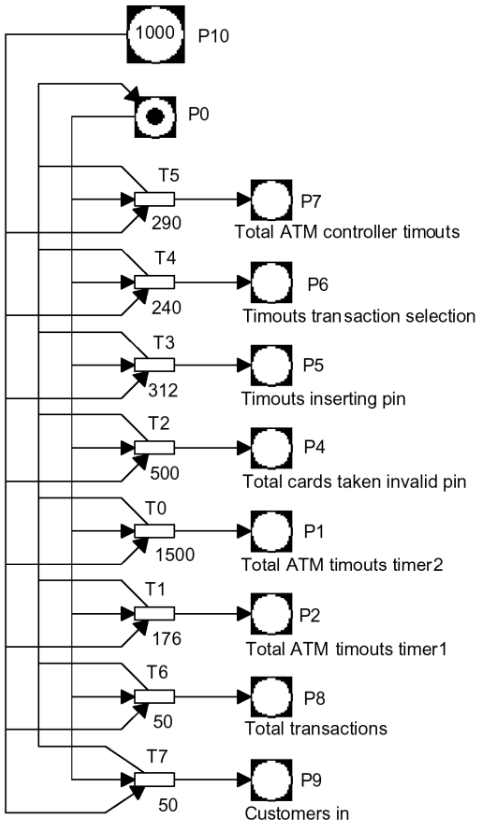

Fig. 2. A simple T.S.Staines model

2.2 NEW DEFINE PLACE

For teller unit we need to various new components. This components design in Petri nets. One of them is new defining place that must be having below characteristics:

i) Ability to store positive token

ii) Ability to store negative token

In basic Petri nets all places had positive or zero token ($M \geq 0$). If number of token at places reached under zero (be negative), place of this token was inactive. But new define places must be have negative token (bidirectional token). For negative place can be used mode in [7]. But negative place don't allow for positive token. HPsim or other simulators usually can't work for bidirectional token. Fig. 3. Shows a basic Petri nets place and transitions. In this nets P1 haven't any token, so T2 is disabling, but P0 have 1 token therefore T1 is enable.

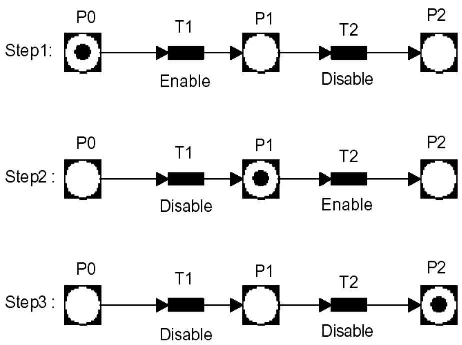

Fig. 3. A simple basic Petri nets

Fig 4. shows a Petri nets with bidirectional place and basic transitions .in this nets P0 haven't any token, so T1 is disable , but ±P1 have 1 token therefore T2 is enable.

In fig 4. T2 is always enabling, but T1 is disabling if P0's token be zero. Because HPsim can't simulate bidirectional token place, we use of positive token place with offset token e.g. token's place set to 10000 (as offset), at the end of simulate if token number for that place be less than 10000 (e.g. 9500) say token is negative (9500 – 10000 = -500) and if be more than 10000 (e.g. 10270) say token is positive (10270 – 10000 = 270)

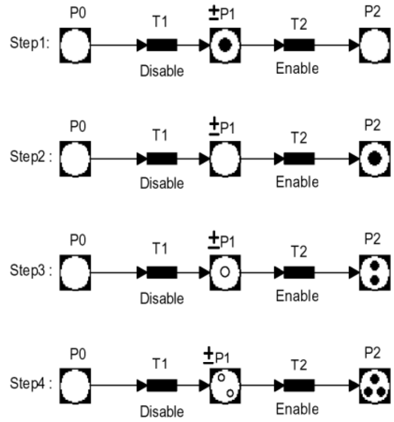

Fig. 4. A simple bidirectional Petri nets

2.3 RANDOM NUMBERS

At this work for payer unit we need to Petri net model for generate random number. Fig. 5. shows this model. T2 gets constant time firing delay, but T1 gets a firing time interval 5 – 10s, so token number of P1 at the end of simulation is a random number Between 5 – 10s.

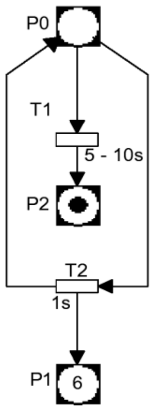

Fig. 5. A Petri net for generate random

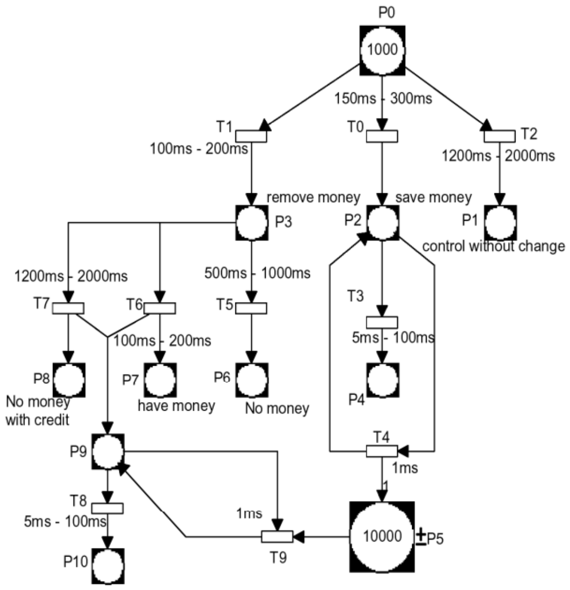

Fig.7. a Petri net for Teller unit

2.4 CHOICE POINTS

Using the idea in Fig. 6 it is possible to add a choice point for the Petri Net. In Fig. 6. There are 0.3333 probability the T 1, T 2 or T 3 fires. Thus we use conflict or choice to simulate random conditions. This design can use for not equal probability.

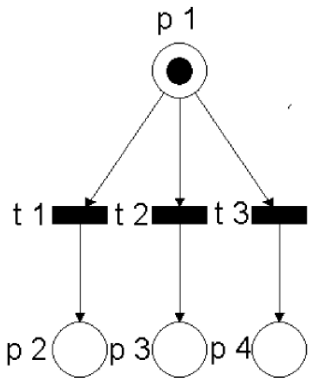

Fig. 6. A Choice points

2.4 FINAL SYSTEM

After each component worked successfully as planned the complete system was assembled. Assembly was simply done by using connection channels. Basically and bidirectional places and transitions were used instead of the dummy places and transitions used for testing. Counters (places) were added to this to be able to get performance results from the Petri Net being used. Simulation was used to derive performance results this implies that the final model needed to be analyzed in detail. Refer to Fig. 7. The functioning of the model was successful.

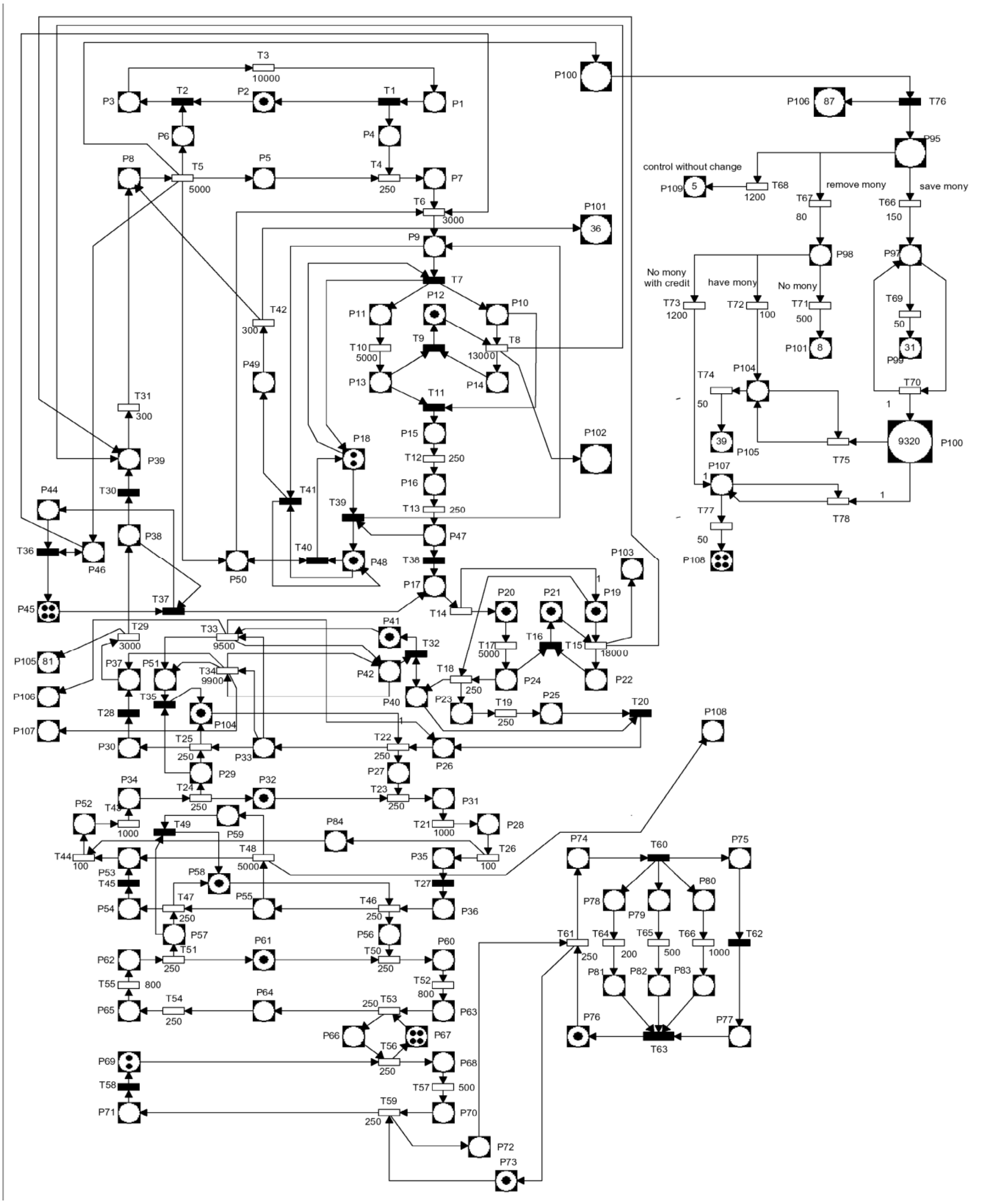

Fig. 8. Complete system timed Petri net consisting of an ATM, ATM controller and Card authorization and money Teller unit

3. RESULTS

Fault tolerant and recovery mechanisms that function correctly have been created. These mechanisms are important for complex systems being a hybrid of hardware and software parts. Timing and performance analysis can be obtained from the models. The final model with the counter places comes close to a generalized stochastic Petri Net which is a special class of a Timed Petri Net .There is also an element of non-determinism because of conflicting transitions or transitions enabled simultaneously.

Table.2. Processing results for 8 runs of the complete system Teller unit

Run No	Coffer	Customers for save Money	Customers for remove Money	Customers No change
1	-994	27	36	11
2	-1428	27	37	11
3	-1484	31	43	12
4	-1217	29	38	10
5	-729	30	36	12
6	-321	29	37	10
7	-1196	26	37	11
8	-1482	27	38	13

The data in Table 2 consists of a summary of 8 simulation runs of one hour each for the model in Fig. 8. Detailed performance analysis was carried out on the model. The values obtained are also quite close to a real scenario e.g. in 1 hour use the system. This is quite realistic. The final results appear in Fig. 9. By changing the time values of the transitions it is possible to achieve totally different results as desired.

4. CONCLUSION

From the results obtained it is shown that added complex mechanisms have been successfully added to normal Timed Petri Nets. It is noticeable that even though the system modeled here initially looks simple there are many intricacies that are often overlooked.

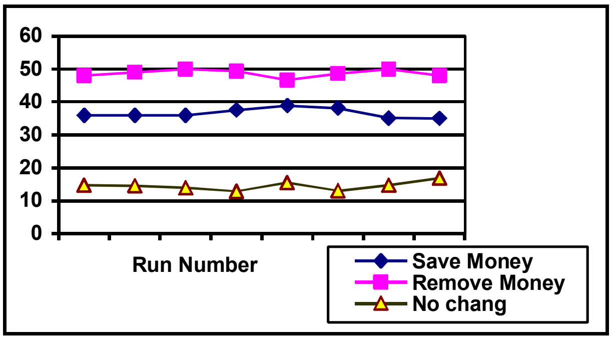

Figure 9. Performance results for the full model

It is possible to refine the Petri Nets developed using reduction methods to simplify the model. These reduction methods enable the Petri Net developed to be simplified in detail. The problem is that many simulation tools do not support higher level constructs and new define places (as bidirectional place). This is why many of these constructs were not used here. These types of diagrams could be used for detailed analysis and design of complex information systems with synchronous and synchronous message passing. These diagrams help to identify possible system. There are also drawbacks with Petri Nets. If the system is complex it is possible to have too many states to control. With new software for Petri net simulator we will can design simplify model with high realistic.

5. REFERENCES

[1] T. S. Staines "Using a Timed Petri Net (TPN) to Model a Bank ATM", Proceedings of the 13th Annual IEEE International Symposium on Engineering of Computer Based Systems (ECBS'06)

[2] G. Booch, "The Unified Modeling Language user Guide, Reading, Mass", Addison-Wesley, 1998.

[3] J. Carlson, "Languages and Methods for Specifying Real-Time Systems", MRTC report, Malardalen Real-Time Research Centre, Malardalen University, Aug 2002.

[4] B.R. Haverkort, "Performance Evaluation of Polling-Based Communication Systems using SPNs" , Lecture Notes In Computer Science 1605, Applications of Petri Nets to Communication Networks, 1999.

[5] Z. Gu and K.G. Shin, "Analysis of Event-Driven Real-Time Systems with Time Petri Nets", IFIP World Computer Congress 2000, Stream 7 on Distributed, Parallel Embedded systems, University of Michigan, 2000.

[6] C.M. Woodside, "A Three-View Model for Performance Engineering of Concurrent Software", IEEE Trans. on Software Engineering, Vol 21 No 9, Sep 1995, pp.754-767

[7] T. E. Lee and S. H. Park, "An Extended Event Graph with Negative Places and Tokens for Time Window Constraints", IEEE Trans. on Automation since and Engineering, VOL. 2, NO. 4, OCT. 2005 pp.319-332