Bi-directional transition nets

Cite as: AIP Conference Proceedings **1836**, 020028 (2017); https://doi.org/10.1063/1.4981968
Published Online: 05 June 2017

Anthony Spiteri Staines

Bi-Directional Transition Nets

Anthony Spiteri Staines

Department of Computer Information Systems, Faculty of ICT
University of Malta
Email: toni_staines@yahoo.com

Abstract. Ordinary Petri nets are forward directed transition systems. Modern transition systems events and event flows are reversible. Hence modeling structures that reflect this are important. The creation of a bi-directional Petri net extends the modeling power of Petri nets. This work presents the successful implementation of a bi-directional transition net. Some toy examples in comparison to Petri nets are given showing the increased modeling power in a compacted form. The results show some interesting findings on how the expressive power of these structures has been increased.

INTRODUCTION

System representation is an interesting and intriguing area that requires considerable techniques and modeling disciplines. In the last decades substantial work and innovative changes have taken place in technologies and architectures of computer systems [1]-[5] and their respective networking. Computer technologies have evolved and developed many new processes and formal or informal systems.

Abstraction is important for representing and comprehending system behavior. Bi-directional flows/arcs give a better representation of system behavior [6]. Different types of structures and models have been brought along to represent these systems. Some of the structures are represented using formal methods and models and informal techniques also. Some models have representation that is good for visualisation.

An interesting area of system modeling is that of symbolic modeling used for system representation. Symbolic modeling is how to computationally represent knowledge [7],[8]. Symbolic architectures offer advantages for human understanding of a system. In this work by symbolic modeling it is intended how to symbolically represent computer processes and architectures in a way that they are better understood by humans. As an idea, symbolic modeling is the use of visual constructs or notations that supply information to different stakeholders [9],[10]. Symbolic modeling here implies that large systems are simplified and reduced. I.e. the state space is compacted through abstraction rendering the model more compact [13]-[15].

Concurrency, causality, conflict and confusion are not so simple to represent and define [11]. For this reason authors have preferred to use reduced or restricted net structures like elementary nets.

In this work the normal view of Petri nets or similar types of networks [12] will be to go against the normal tradition of having Petri nets as being forward ordering transition systems. The behavior of a transition is more complex to represent than is normally considered.

RELATED WORKS: TRANSITION SYSTEMS

Transitions are the founding blocks of all system major processes. Transitions represent events that are taking place or are bound to take place in the future. Past events are also part of the scenario. Events can be composed of a number of transitions that occur in some causal ordering sequence. Different notations have been created to model these types of behavior. The simplest form of diagrams for representing transitions is the state transition diagram or state transition graph. To this transition diagram, different notations and elements have been added making it more complex and capable of representing more detail. In software modeling and software engineering other notations have been created like automata, UML activity diagrams, transition systems and Petri nets.

Applied Mathematics and Computer Science
AIP Conf. Proc. 1836, 020028-1–020028-6; doi: 10.1063/1.4981968
Published by AIP Publishing. 978-0-7354-1506-5/$30.00

Pictorially a transition system can be represented as an edge labelled digraph with a given root [9],[10]. Hence this type of system can be called a forward directed transition system, implying that the future states of the system are the result of the current states.

Formally a transition system is a pair (S,->) where S is a finite set of states, S= s1,s2,s3,…..,sn and -> is a finite set of transitions t1,t2,…..,tn. A transition from a state s1 to s2 i.e. (s1,s2) is given as E ->, s1 ->s2. A labelled transition system (LTS) is a tuple (S,A,->). The edge from s1 to s2 is written s1 –a-> s2. I.e. the edge is given a label [9],[10].

Transition systems (TS) are used to describe the potential behavior of discrete systems. TS consists of states and transitions between states. System labelling can be simple or complex. Simulation pre-order represents the relationship between state transition systems where the moves of another system are intuitively matched.

Petri nets are bi-partite graphs with special added features that can model concurrency, mutual exclusion, sequence and control in communicating systems that exhibit discrete behavior [11]. Petri nets can be considered to be forward directed transition systems that are not normally reversible.

MOTIVATION AND PROBLEM DEFINITION

A Petri net can be structured in such a way as to reverse its state, however this is normally done by adding more places, arcs and transitions. Reversing the state implies that deterministic behavior is still being used. A new network topology or a Petri net with undirected edges connected to nodes is required. This implies that transition firing can happen both ways i.e. bi-directionally. Modeling power is drastically increased and non-determinism principles are introduced. Behavior that is not allowed to take place in the definition of classic Petri nets is allowed here. Obviously many new issues are created and some form of control principles or rules might have to be defined. The advantage of this approach is that the modeling power of Petri nets is extended and reflects in a more precise way what takes place in modern systems where normally it makes sense to be able to reverse transitions. As a simple example, in many modern autonomic or autonomous control systems it is possible to have state reversal as an automatic or reset process. Considerable research has been carried out in the direction of transition systems [7]-[9]. These structures are based on digraphs and are suitable for visualization.

Symbolic modeling is imperative for understanding and representing system structures. Visualization techniques are useful for representing system structures like networks. Bi-directional transitions serve to explain in better detail the real scenario. However, it must also be considered that having transitions that can fire bi-directionally definitely increase the undecidedness in the system. Some reasons for bi-directionality in transition firing are: i) it is more realistic, ii) shortcut modeling can be used, iii) weak association and weak controls can be introduced, iv) more realism of what happens in the real world is introduced.

PROPOSED SOLUTION

The proposed solution here a prima facie seems to be very simple. Instead of using directed arcs as in normal Petri nets the direction of the arcs is simply removed creating undirected arcs.

However, this creates a number of issues that need to be addressed. The formal solution can be given in future works. This work outlines the basic idea of doing this. By removing the direction of the arcs it is implied that tokens in the net can travel in both directions. This creates a structure that is more complex and abstract than a normal Petri net. The concept of undecidedness is introduced and the normal orderly or linear firing sequence that is found in Petri nets is no longer there. Even the normal firing routine in Petri nets is changed.

This type of structure will closely imitate what happens in the real world, where behavior is not necessarily ordered. The bi-directional transition net is suitable for i) static and ii) dynamic modeling. The latter is more difficult to control and will require some specific rules according to the case being modelled.

IMPLEMENTATION

The following general rules will serve to give a brief overview of the implementation of the BDTN. i) Places can serve as inputs or outputs. Alternatively they can be called channels or stores. ii) For transition firing there have to be a sufficient amount of tokens in the input places. However, some input places can be empty because firing is

controlled by the transition and not the arcs. iii) There are various possibilities for output once a transition fires. One place can be left empty and the tokens moved into another place. This is different from traditional Petri nets where normally the output places cannot be left empty after transition firing. iv) Additional rules might need to be added independently to a model for a particular scenario. This would be a scenario related rule. v) Some form of ordering in firing might be necessary.

SOME TOY EXAMPLES

A Simple Web Login Process

Fig. 1 clearly illustrates the concept of the BDTN in comparison to a typical Petri net. The idea of a web login process is introduced and compared. In fig. 1 a) the Petri net shows that at each stage of the process e.g. at the *pwd entered* stage it is possible to roll back to a previous state, this is achieved by the extra transitions that have been added. In fig. 1 b) the BDTN shows the same functionality of the Petri net but at each stage connecting arcs have no direction, implying that the tokens can flow bi-directionally.

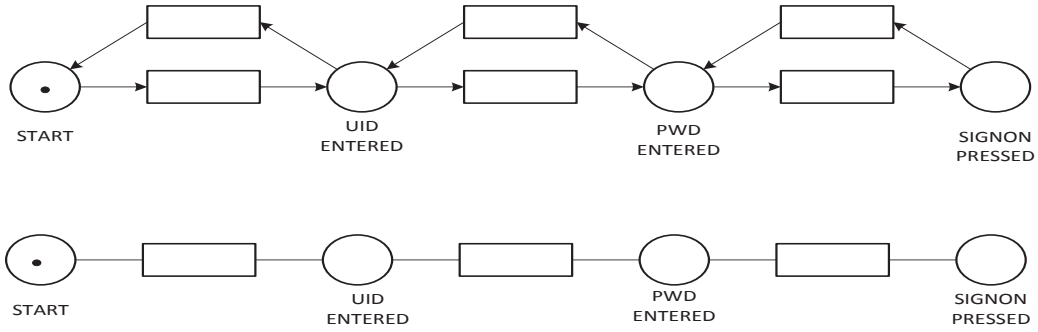

FIGURE 1. *A Simple Web Login Process a) Petri Net Model and b) Bi-Directional Model*

Static Comparison of a Petri Net with a Bi-Directional Net

Fig. 2 shows how the BDTN compares with a normal Petri net. In a normal Petri net arcs are always directed and are either input or output arcs. On the other hand this paper is proposing a net structure where the arcs do not have a specific direction or are unidirectional. The overall structure still looks like a normal Petri net from the static point of view. However, the BDTN's dynamic functionality will be significantly different than that of the Petri net.

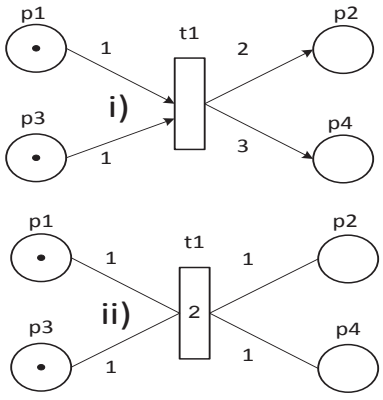

FIGURE 2. i) *A Simple Petri Net vs ii) A Bi-Directional Net*

Dynamic Operations of an Elementary Bi-Directional Net

Fig. 3 shows how the dynamic operations for the BDTN given in fig. 2 ii). i.e. fig. 2 shows all the possibilities that are available when firing the main transition T1. Note that this is not a given sequence of how the outputs occur and all places have a restriction of one and the arcs can carry only one single token. The transition requires a minimum input of two tokens for firing. This is clearly indicated by the number two in the transition. There is no given temporal order how this firing will affect the output. I.e. the output and the input can happen in any order. The output can become the input for the next transition firing. Hence, the structure in fig. 2 ii) is a live structure that is continuously enabled. However fig. 3 shows all the possible six states of the system which are reachable. The structure is a live structure without deadlock.

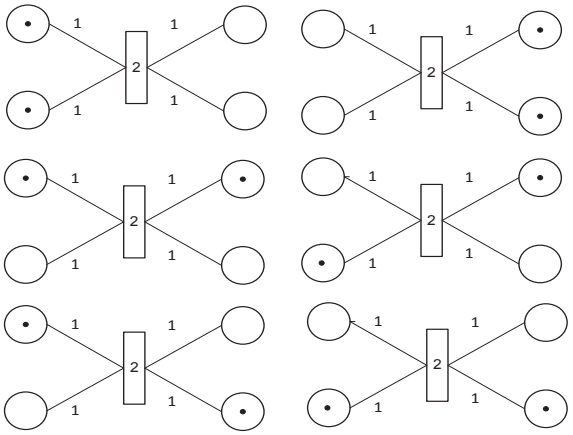

FIGURE 3. *Possible Markings/States for an Elementary Bi-Directional Net*

A Simplified Version with a Single Token

Fig. 4 shows a simpler version of fig. 3. This has been obtained by reducing the token to one and the transition requires only a single token for firing. Clearly the system has less states. These are just four states. It is clearly indicated that firing is not associated with a particular order.

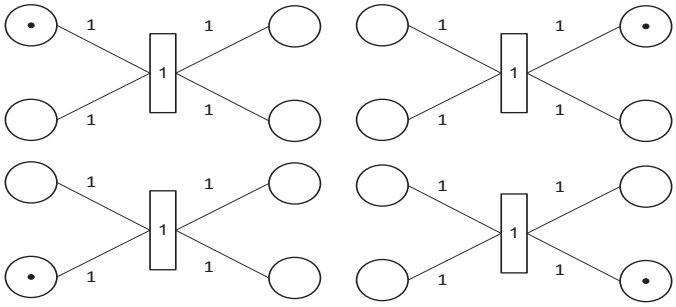

FIGURE 4. *Simplified Version with only One Token*

A More Complex Bi-Directional Net

Fig. 5 depicts a more complex BDTN. In this example transition t1 and t2 require a minimum of two tokens for firing. The tokens can be present in any of the places connecting to t1 and t2. E.g. t1 is enabled and can fire as long as two tokens are available together from any given combination p1,p2,p3 and p4. In fig. 5 t1 is enabled because

there is one token in p1 and one token in p2 which give the two tokens required by t1 for firing. If t1 fires then the possibilities listed in fig.3 are all available. Transition t2 has been restricted. I.e. it can fire only if one token is present in p3 and one token in p4 or two tokens are present in p5. So t2 functions like a two way switch. This can be done because the place capacity of p5 is two and the connecting undirected arc from t2 to p5 and vice-versa has a weight of two implying that either two tokens are outputted or inputted in a single instance of firing t2. Fig. 6 indicates the presence of choice in this net. In fig. 6 having one token in p3 and one token in p4 and the concept of bi-directionality implies that both transition t1 and transition t2 are simultaneously enabled. But only one can fire because firing one automatically will disable the other.

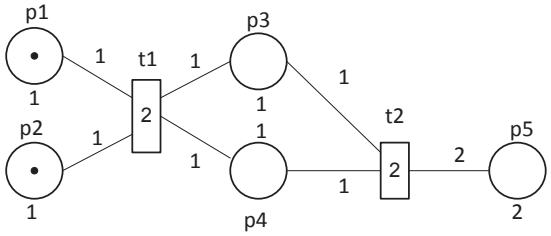

FIGURE 5. *A Complex Bi-Directional Net*

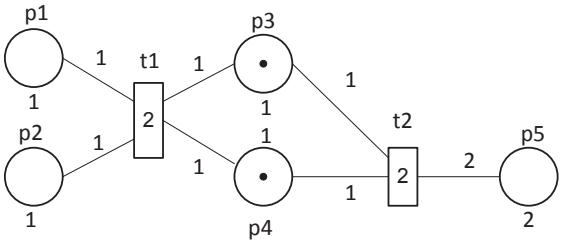

FIGURE 6. *Choice in the Bi-Directional Net*

A Directed Petri Net Circuit vs a Bi-Directional Net

Fig. 7 shows a normal Petri net circuit and a BDTN. Even though structurally they look similar there are big differences in the operations that are possible. In the Petri net tokens can flow only in one direction. On the other hand in the BDTN the token flow can be in both directions. So the BDTN serves as a dual circuit.

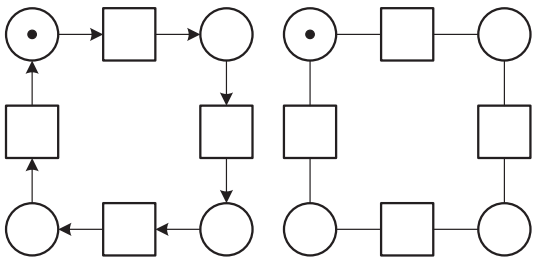

FIGURE 7. *A Normal Petri Net Circuit and a Bi-Directional Net Circuit*

RESULTS AND FINDINGS

This work has concentrated on giving a brief overview on a new type of Petri net like topology which has been called a BDTN. In essence this structure retains many of the main structural properties found in general Petri net classes. However the behavior and firing is not similar. It is obvious that the BDTNs offer more detail than Petri nets

at the expense of having complex behavior that might not always be controllable. If the BDTNs are properly constructed and restricted it is possible to achieve better control. This is indicated in fig. 6 where the possible firing of the net has been restricted by using place capacity restriction and arc weights. The simple web login process shows the usefulness of the BDTN as a static and dynamic modeling notation for very simple processes that are serially linked together. Fig. 2 shows that even though the structure looks simple, just by increasing the number of tokens in the places it is possible to generate complexity. This can be seen when fig. 3 is compared to fig. 4. Fig. 7 compares a normal Petri net with a BDTN. The bi-directional is reversible at every instance and hence it expresses more behavior than the Petri net. To represent the same behavior using a Petri net would require adding more arcs and transitions. Hence the result of this is that the BDTN is shown to be more compact.

The main results of this work have shown that i) it is possible to construct a BDTN, ii) the BDTN can capture more detail than a Petri net, iii) the BDTN will require some additional form of restriction or control.

CONCLUSIONS

This work shows the successful implementation of a BDTN. From the structural point of view the BDTN looks similar to a Petri net, however the fundamental operations and transition firing differ. This work is incomplete in the sense that just a very trivial idea has been presented. Proper rules for firing have not yet been established. It is up to the user to interpret and set the firing rules for this structure thus allowing a lot of room for more possibilities.

REFERENCES

1. A. Spiteri Staines, *A Colored Petri Net for the France-Paris Metro*, (NAUN, International Journal of Computers, Issue 2,Vol. 6., 2012), pp. 111-118.
2. T. Spiteri Staines and F. Neri, *A Matrix Transition Oriented Net for Modeling Distributed Complex Computer and Communication Systems*, (WSEAS Transactions on Systems, Vol 13, 2014), pp. 12-22.
3. T. Spiteri Staines, *Implementing a Matrix Vector Transition Net*, (British Journal of Mathematics & Computer Science, ISSN: 2231-0851,Vol.: 4, Issue.: 14, 2014), pp. 1921-1940.
4. A. Spiteri Staines, *Some Fundamental Properties of Petri Nets*, (International Journal of Electronics Communication and Computer Engineering, IJECCE, vol.4, Issue 3, 2013),pp. 1103-1109.
5. K. van Hee, *Information Systems: A Formal Approach*, (Cambridge Univ. Press, 2009).
6. G.-C. Yang, "Distributed System Modeling with Bidirectional Petri Nets", *in Proc. Of the 'Computer Systems and Software Engineering conf.. (CompEuro '92)*, (IEEE,1992), pp.401- 405.
7. D. Kleyko, E. Osipov, "On bidirectional transitions between localist and distributed representations: The case of common substrings search using Vector Symbolic Architecture", *5th Annual Int. Conf. on Biologically Inspired Cognitive Architectures, Procedia Comp. Science,* (BICA, 2014), pp 104-113.
8. T.D. Kelly, *Symbolic and Sub-Symbolic Representations in Computational Models of Human Cognition*, Theory& Psychology, (Sage Publications: Vol. 13, No. 6, 2003), pp. 847-860.
9. F. van Ham, H. Van de Wetering And J.J. van Wijk, *Interactive Visualization of State Transition Systems*, (Transactions on Visualization and Computer Graphics, Vol. 8, No.3, IEEE, 2002), pp. 1- 11.
10. J. Osis, and E. Asnina, *Topological Modeling for Model-Driven Domain Analysis and Software Development: Functions and Architectures*, (Model-Driven Domain Analysis and Software Development: Architectures and Functions, 2010), pp. 15-39.
11. T. Murata, *Petri Nets: Properties, Analysis and Applications*, (Proc. Of the IEEE, Vol 74 issue 4, IEEE, 1989), pp.541-89.
12. M. Bouhalouane, S. Larbi And H. Haffaf, "Combining Bond Graphs and Petri Nets Formalisms for Modeling Hybrid Dynamic Systems", *10th Int. Conf. on Future Networks and Communication,* (Science Direct, Procedia Computer Science, Elsevier, 2015), pp. 252-259.
13. H. Kaindl And J.M. Carroll, *Symbolic Modeling in Practice*, (Communications of the ACM, vol. 42, No 1, 1999), pp. 28-37.
14. D. Ross, *Structured Analysis: A Language for Communicating Ideas*, (IEEE Trans. Softw. Eng. Vol.3, No 1,1977)
15. R.A. Kremer, B.R. Gaines, "Embedded Interactive Concept Maps in Web Documents", *In Proc. Of WebNet96, H. Maurer, Ed.* (Charlottesville, VA, 1996), pp. 273-280.