Modelling and Verification of Cyber Physical System Using Timed Petri Net

George Enekwa[1]

Contents

[1] Izuchukwu-george.enekwa@stud.hshl.de

Abstract: Development processes, networking and the evaluation of models are becoming more demanding, especially in the areas requiring, high verification, safety and security. These are being studied in many scientific and technological fields and with the blooming of Cyber Physical Systems (CPSs), distributed systems are becoming more reliabl and the usage of a simple and more definitive modelling strategy is becoming increasingly crucial. In this paper, a timed Petri nets based strategy enabling behavioral modeling and performance analysis of Cyber-Physical-Social Systems (CPSSs) is presented. This also addresses uncertain scenarios where the social aspect is also having a significant impact on the functioning of these systems. Petri nets models, included with dynamic time dependencies that associated with transitions, are applied in a case study, and corroborated as a potential tool for modelling and analysis of these kind of systems.

1 Motivation

The notion of Cyber-Physical Systems emerged in recent years as a result of the integration of computer or cybernetic systems with physical systems (CPSs). CPSs are heterogeneous entities that span the hardware and software, sensors and actuators, and other domains. Such systems must be extremely adaptive and flexible to respond in non-deterministic and changing situations with acceptable performance because they are typically used for dynamically changing purposes. Eventually, the human Studying perspective as a whole and crucial component must be necessary to carefully design such systems. Thus, these systems must be carefully evaluated because they are mostly deployed in security-critical applications where their failures can have serious consequences. Public services, smart factories, smart healthcare, and smart cities can all be categorized as a few common CPSS applications. Their modeling is more challenging due to the CPSSs' inherent complexity as well as the connections and interactions between system components. Concurrency, synchronization, distributed, real-time, discrete, and continuous aspects are among the characteristics of these kinds of systems. With regard to smart cities, a wide range of connected topics should be taken into account and examined in light of the high system performance, including intelligent traffic management and intelligent transportation systems, to name a few. In order to represent CPSSs in uncertain environments, this study introduces a correct formalization, utilizing an intelligent traffic management system as a validation example. Petri nets (PN) are well-suited to deal with the challenges of CPSSs among modeling formalisms suitable for use in the specification, analysis, and implementation of CPSSs. They support a model-based development strategy, including component design, orchestration of components, as well as component and overall performance evaluation. In this study, non-autonomous Petri net modeling is utilized for the definition, analysis, and implementation of CPSSs. The PN model's inherent properties were enhanced by the addition of dynamic time related with the development of the model, as suggested in the section below. The proposals are validated using an application example from a traffic light management system for intersections, where the arrival rate of automobiles are taken into consideration to constraint the behavior affecting the system's performance.

2 Systems Overview

2.1 Cyber Physical Systems:

A cyber physical system (CPS) is an integration of computing, communication with monitoring or/and control of things in the physical environment, according to a popular description given for the concept. A key component of the theory combining computing, communication, and control is information. Information may come from several sources, including societies, human beings, and physical objects like sensors and actuators, or several sources, including networks that monitor and regulate the physical processes, typically with feedback loops where the physical processes influence computation, human operators, and embedded computers as depicted in Fig. 1. CPS are frequently operated by human operators, so human factors need to be incorporated into the design of such systems. They are heterogeneous, concurrent, and time sensitive, so modeling them is challenging. In order to model CPSs, both qualitative and quantitative models are needed since they are discrete and dynamic in nature. Petri nets are a popular formalisation of CPS.

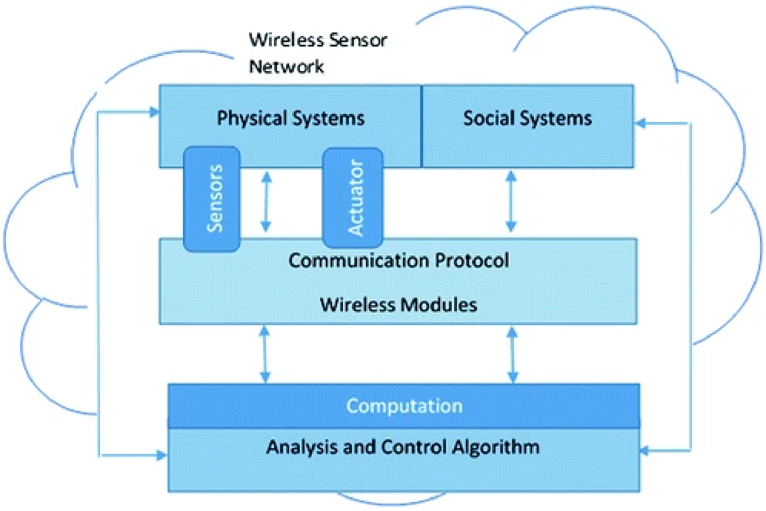

Fig. 1: Cyber Physical System.

By applying modeling formalisms, like Petri nets, to different phases of development, such as specification, and implementation, as well as allowing a-priori verification of properties and anticipating the impact of failures or misconduct, systems' resilience can be improved. The adoption of a model-based development attitude, such as the one proposed in this paper, can contribute to improving confidence in the functioning correctness of real-time operations, survivable during attacks, and fault tolerance.

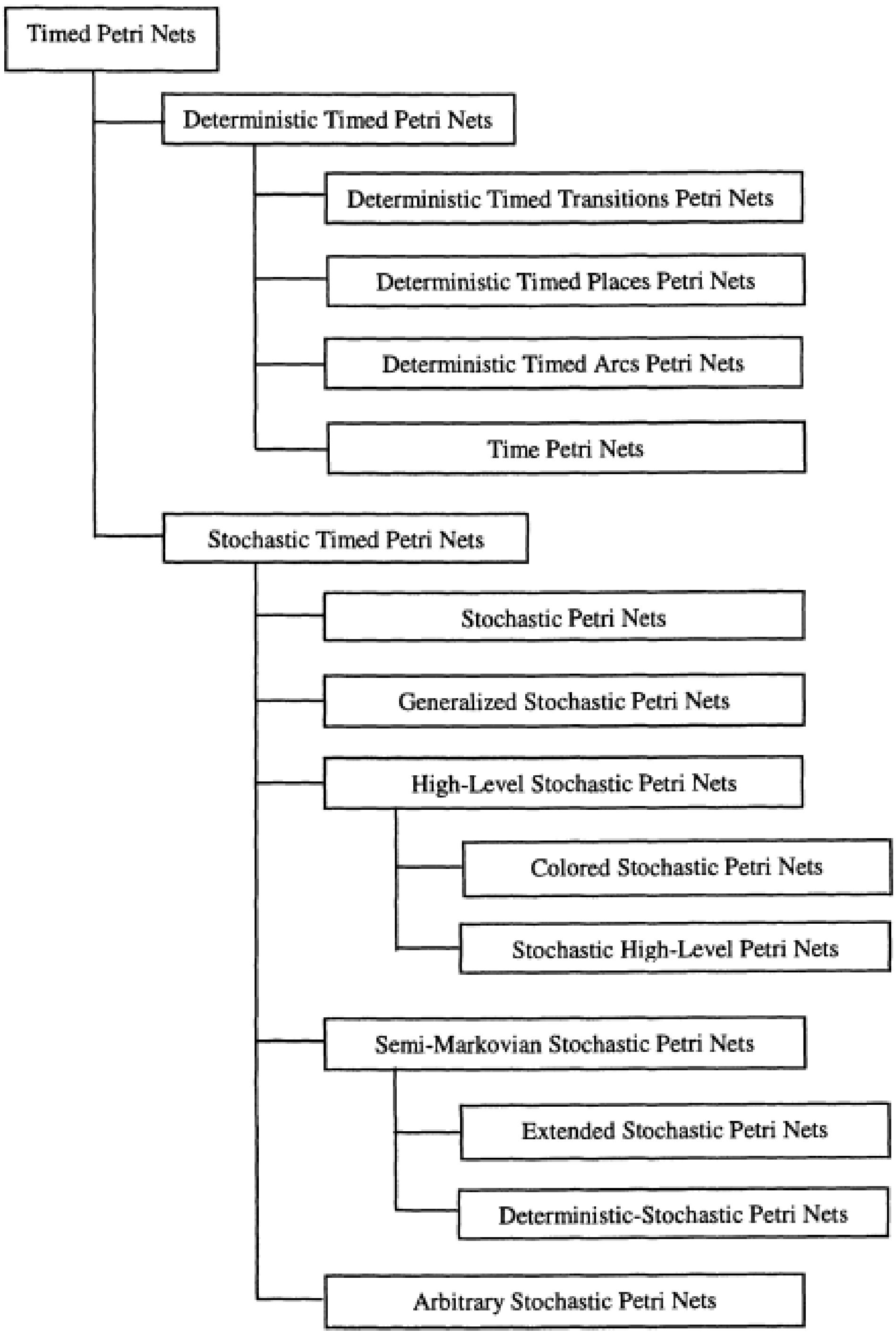

Fig. 2: Classification of timed Petri nets.

2.2 Petri Nets:

The Petri net (PN) provides a clear means of presenting simulation and control logic for a wide variety of discrete event systems.As shown in [HCO8] and [Hu12], they are therefore suitable for designing traffic control systems. Despite this, they cannot determine the exact time of transition firing without extending the time dimension.Thus, they can only be used to analyze the functional or qualitative behavior of the systems. Time Petri nets are deployed in this case to enhance the capability. In recent years, TPNs have been successfully used to model railway level crossings and urban traffic control systems. Additionally, timed coloured Petri nets (TCPNs) are used to model complex systems visually.

2.3 Timed Petri Net:

A Time Petri net can be obtained from a Petri Net by associating two dates with each transition. Assuming that t became active for the end time at date , then t cannot fire (cannot be taken) before the date + min and must fire no later than the date + max, except if firing another transition disables t before then. As time Petri nets naturally express specifications ïn delays,"by making explicit when an action begins and ends, they can also express specifications ïn durations."They have broad applicability. A timed Petri Net can be used as both a logical and a quantitative model. These models allow the same language to be used for specification, validation, and estimation of functional/logical properties (e.g., the number of deadlocks) and performance properties (e.g., average waiting times). As seen from Figure 2, There are two types of timed Petri Nets: deterministic timed petri nets, in which every transition, place, or directed arc has a deterministic firing time or time interval, and stochastic timed petri Nets, in which each transition has a random firing time. In this study, we will focus more on the deterministic timed Petri Net family. It consists of Deterministic Timed Transitions Petri Nets that have each transition associated with a specific firing time, deterministic Timed Places Petri Nets that have a specific firing time for each place, deterministic Timed Arcs Petri Nets whose directed arcs are associated with deterministic firing times, and Time Petri Nets whose transitions are associated with deterministic firing times. To verify and model a Cyber Physical System, a traffic light case study was implemented in this study using deterministic timed petri net.

3 Introducing a Traffic Light Control System

Taking a look at traffic management, a common problem that affects millions of people on a daily basis,its relation to time, was the focus of this study. An intersection traffic light system consists of a physical controller and a computation section. Furthermore, the performance of the system is also affected by social factors. Modeling of intelligent traffic light control systems involves considering several variables that constrain the time management of the system. Here are some of the considered variables; Number of queued vehicles, pedestrians and etc. It is the goal of an intelligent traffic system to reduce queuing times and improve traffic flow by controlling the volume of traffic. For this reason, it is essential to design a model that will yield efficient and safe operations under the current conditions.There are several operation modes that have previously been studied and are considered (night, day, rush hours, etc.), each with predefined sequences and temporal behavior (fixed or nearly fixed periods of time). This paper proposes a modelled system that controls vehicles at the intersection based on time periods.

3.1 Modelling Crossroad Traffic Light Control Systems using Timed Petri Net

On the basis of the above discussion, this section illustrates a crossroad traffic light control system using TPNs. A general traffic system with two-phase traffic lights can be seen in Figure 3(a). Some significant regulations are required when taking vehicle safety into account.

- If all of the traffic signal lights are in the red condition, a traffic light control system can be activated.
- There cannot be more than one green light on at a time, and
- The traffic signal alternates between red, green, and yellow.

The TPNs in Fig. 3 accurately simulates the overall operation of the system. The system consists of three traffic signal lights, namely red (R), yellow (Y), and green (G). Northward, southward, westward, and eastward directions are symbolized as sn, ns, ew, and we, respectively. The TPNs model is built using two different sorts of locations. Places of Type I consist of Rns, Yns and Gns, corresponding to R, Y, and G traffic lights.The type II ones are Rwe, Ywe, and Gwe.It is possible to derive the TPN transitions of the system based on the above described two-phase.

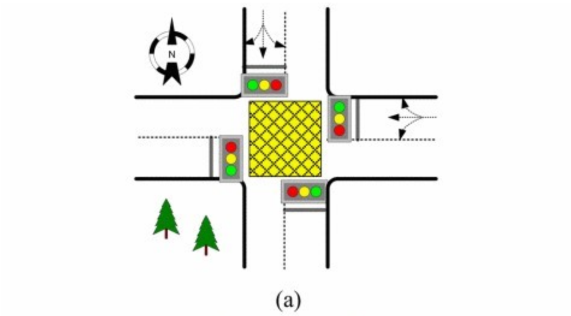

Fig. 3: (a)A two-phase traffic light control system.

3.2 Analysis and Simulation

In this section, the analysis and simulation of the system was made, which were true in accordance with the design specification, as seen in the figures below. Figure 3. shows the system state machine diagram with the supposed assigned transition time intervals.

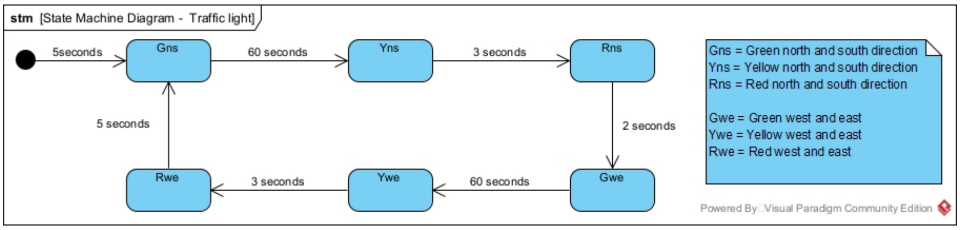

Fig. 4: State machine diagram.

The system in Figure 4. shows the flow of the system in respect to the specified time intervals listed in section 4. There are six transitions (t1, t2,t3, t4, t5, and t6) with deterministic firing times; red is the signal time in direction ns (we), green is the green signal time, and yellow is yellow for the signal time in direction ns (we). A proper analysis and test of the system was done using Tapaal, the system simulates repeatedly as required with no deadlocks, also structurally bounded and live.

The traffic light system's TPNs model can be built in accordance with the specifications as seen in Figure 4. The starting state of the traffic control system model is depicted in Figure 4 as the red states.

From Figure 5, when t4 is fired, a token is moved into Green-ns and P2, respectively, after 5 seconds. The green light is now on, allowing northbound and southbound traffic to proceed through the crossing. The green light should then be turned off after the length of t1.The green light has apparently been on for 60 seconds. And for the next transition, as t0 lasts for 3 seconds, the yellow light stays on for 3 seconds. The token is then transferred back into Red-ns. In 2 seconds (following firing of t5), a token in Red-we moves into Green-we. The signal light T3 can fire when Green-we becomes green, the Green-we stays on for 60 seconds and then the T3 fires in, Due to the duration of t2, the yellow light is on for 3 seconds.The token is finally placed back into Red-we.

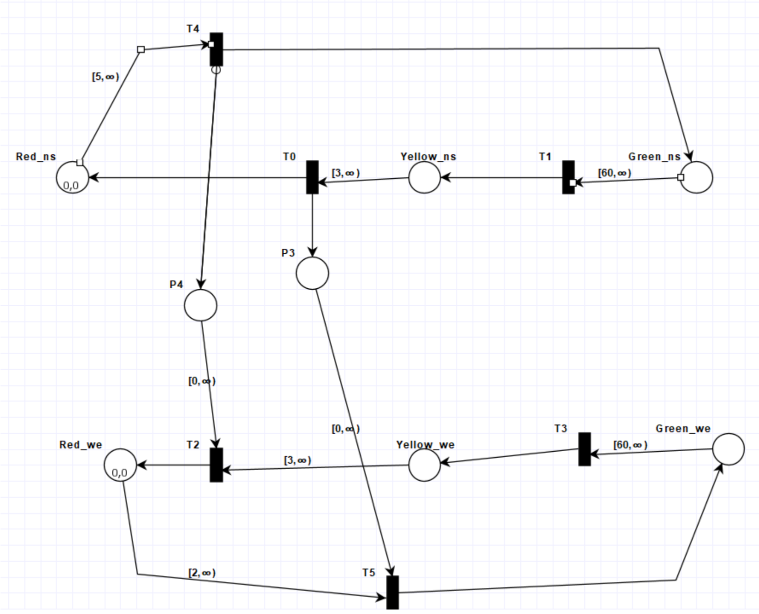

Fig. 5: Traffic light timed petri net model.

4 Verification and validation of the systems Timed Petri Net model

As a first step, we define a marking function, which assigns a finite multiset of nonnegative real numbers to each place. Real numbers are used to represent the ages of tokens at a given location. The ages of tokens should also respect the invariant of the place where the token is located. A marked TPN consists of a pair (N, M0) where N is a TPN and M0 is an initial marking on N where all tokens have the same age.

Definition 3 (Enabledness). *Let N = (P, T,IA, OA, Transport,Inhib,Inv) be a TAPN. We say that a transition t T is enabled in a marking M by tokens if*

- *for all input arcs except the inhibitor arcs there is a token in the input place with an age satisfying the age guard of the arc, i.e.*
- *for all inhibitor arcs there is no token in the input place of the arc with an age satisfying the age guard of the arcs respectively, i.e.*
- *for all input arcs and output arcs which constitute a transport arc the age of the input token must be equal to the age of the output token and satisfy the invariant of the output place, i.e.*
- *for all output arcs that are not part of a transport arc the age of the output token is 0, i.e.*

Definition 4 (Firing Rule). *Let N = (P, T,IA, OA, Transport,Inhib,Inv) be a TAPN, M a marking on N and t T a transition . If t is enabled in the marking M by tokens In and Out then it can fire and produce a marking M*

Definition 5 (Time Delay). *Let N = (P, T,IA, OA, Transport,Inhib,Inv) be a TAPN and M a marking on N. A time delay d R0 is allowed in M if (x + d) Inv(p) for all p P and all x M(p), i.e. by delaying d time units no token violates any of the age invariants. By delaying d time units in M we reach a marking M0*

In TPN there is a timed transition system where states are markings of N, so if there are two markings M and M0 we can reach the marking M0 by firing some transition in M and by delaying the time units in M we can reach the marking M0. That is to say that a marking M0 is reachable from a marking M if M0 is reachable from M.

To properly verify a TPNs model and make sure all the above statements are met, the questions of the Petri net problems like reachability, coverability and boundedness in the TPN context must be checked. The reachability of the basic timed-arc Petri net model is undecidable (with only ordinary arcs and no age invariants). However, coverability, boundedness, and other issues remain decidable for the basic TPN model [Ja11].

For the above Model (Figure 5), I checked if every reachable marking in the net satisfies the given property and if there is a trace on which all markings satisfy the given property. The results were true and successful.

5 Conclusion

In this study, TPNs models for a traffic light level crossing systems are proposed. It is important to note that the presented models make use of the idea of hybrid systems.They can be used for timing analysis.In particular, the transition used is timed. Which is therefore used to model the traffic light behaviour. It is important to note that the emerging scenarios can be accurately identified using the proposed models. Upon investigation, the proposed system model is free of deadlocks, exhibits repetitive behavior, and is structurally bounded and live. The analysis of the models proves so. Based on the validation provided by the application example, Timed Petri nets can be considered a promising formalism for modelling CPSS-type systems.

6 Declaration of Originality

I, Izuchukwu George Enekwa, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

Bibliography

[HC08] Huang, Yisheng; Chung, Ta-Hsiang: Modeling and Analysis of Urban Traffic Lights Control Systems Using Timed CP-nets. J. Inf. Sci. Eng., 24:875–890, 2008.

[Hu12] Huang, Yi-Sheng; Weng, Yi-Shun; Liu, Maw-Yang; Chen, Jun-Ren: Traffic Light Controller for Parallel Railroad Level Crossing Traffic Control Systems Using Deterministic and Stochastic Petri nets. IFAC Proceedings Volumes, 45(6):321–326, 2012. 14th IFAC Symposium on Information Control Problems in Manufacturing.

[Ja11] Jacobsen, Lasse; Jacobsen, Morten; Møller, Mikael; Srba, Jirí: Verification of Timed-Arc Petri Nets. pp. 46–72, 01 2011.

[Jo06] Joseph, A.D.; Beresford, A.R.; Bacon, J.; Cottingham, D.N.; Davies, J.J.; Jones, B.D.; Guo, Haitao; Guan, Wei; Lin, Yong; Song, Houbing; Iftode, L.; Fuchs, S.; Lamprecht, B.; Kyamakya, K.; Fernandez, J.G.; Yelmo Garcia, J.C.; Martin Garcia, Y.S.; de Gracia Santos, J.; Nimesh, M.; Pan, Gang; Wu, Zhaohui; Wu, Qing; Shan, Zhenyu; Sun, Jie; Lu, Jian; Yang, Guoqing; Khan, M.K.; Zhang, Jiashu: Intelligent Transportation Systems. IEEE Pervasive Computing, 5(4):63–67, 2006.

[oSE00] on Systems Engineering, International Council, ed. INCOSE Systems Engineering Handbook, volume 2.0. 2000.

[Wa10] Wang, Fei-Yue: The emergence of intelligent enterprises: From CPS to CPSS. IEEE Intelligent Systems, 25(4):85–88, 2010.

[ZZ94] Zurawski, Richard; Zhou, Meng Chu: Petri Nets and Industrial Applications: A Tutorial. IEEE Transactions on Industrial Electronics, 41(6):567–583, December 1994. Funding Information: Manuscript received Decemher 12, 1993; revised July 22, 1994. Partial financial support was given hy the Center for Manufacturing Systems at the New Jersey Institute of Technology. R. Zurawski is with Laboratory for Robotics k Intelligent Systems. Swiburne University of Technology, Melbourne. Vic. 3122. Australia. M. C . Zhou is with the Laboratory for Discrete Event Systems, Department of Elcctrical and Computer Engineering, New Jcrsey Institute of Technology. Newark, NJ 07102 USA. IEtE Log Number 9405099.

[Wa10] [Jo06] [HC08] [Ja11] [Hu12] [oSE00] [ZZ94]