PIPE PLUGIN FOR HYBRID PETRI NETS VISUAL SIMULATION OF WIRELESS SENSOR NETWORKS

Andrei FURTUNA

Technical University of Moldova

Abstract: *This paper presents an overview of proposed plugin for adding feature of modeling hybrid petri nets for Platform-Independent Petri Net Editor (PIPE), an open-source tool that supports the design and analysis of Generalized Stochastic Petri Net (GSPN) models. Petri nets are a widely used formalism for the analysis of concurrent systems and as such there are a lot of existing tools which allow users to edit, animate and analyze a range of Petri net classes. PIPE 's extensible design enables developers to add functionality via pluggable analysis modules. It also acts as a front-end for a parallel and distributed performance evaluation environment.*

Keywords:*Wireless sensor networks (WSN), hybrid Petri Nets, computer system.*

Introduction

We can see that latest trends in wireless sensor networks (WSN) has touched a wide range of applications of WSNs in military sensing, traffic surveillance, target tracking, environment monitoring, healthcare monitoring, and so on. Here we describe such type advances in WSN and their applications in various fields.

Topology control is one of the fundamental problems in WSNs. It has great importance for prolong lifetime, reducing radio interference, increasing the efficiency of media access control protocols and routing protocols. It also ensures the quality of connectivity & coverage and increase in the network service as well. A significant progress in research can be seen in WSNs topology control. Many topology control algorithms have been developed till date, but problems such as lack of definite and practical algorithm, lack of efficient measurement of network performance and idealness of mathematical model still exist. [1][2]

For modeling and simulating a WSN topology it can be used different tools and formal languages, we have chosen to use Petri Nets (PN) which are a well-accepted formalism for modelling concurrent and distributed systems in various application areas: workflow management, embedded systems, production systems, WSN and traffic control are but a few examples. The main advantages of PNs are their graphical notation, their simple semantics, and the rich theory for analyzing their behaviour.

Performance verification and evaluation are some of techniques widely used to design and analyze different distributed systems and hybrid applications with discrete-continuous processes [3]. Using this techniques design, gaps can be identified at the beginning stage [4, 5, 6, 7]. Thus these problems can be eliminated earlier, and troubleshooting, maintenance and maintenance costs can be significantly reduced.

In spite of their graphical nature, getting an understanding of a complex system just from studying the Petri net model itself is quite hard – if not impossible. In particular, this applies to experts from some application area who, typically, are not experts in PNs. [8]

Platform-Independent Petri Net Editor (PIPE) is a Java-based tool for the construction and analysis of Generalised Stochastic PN (GSPN) models. PIPE began life in 2002/3 as a postgraduate team programming project in the Department of Computing at Imperial College London and has been steadily improved through a number of successive versions, implemented by students at Imperial College London and with input from industry (particularly Intelligent Automation, Inc.) In addition, a branched version with significant improvements to different aspects of functionality (e.g. the addition of inhibitor arcs and fixed-capacity places, and an experimental framework) has also been implemented at the Universitat de les Illes Balears. [9]

In this paper we propose to add the feature for modeling hybrid PNs (HPN) in the above mentioned software. This will facilitate the analysis and visualization of further work in domain of Wireless Sensor Networks, which is one of the top trends from last decades.

In Section 2 we briefly introduce the definition of HPNs. Then we proceed, in Section 3, to present the PIPE tool and the way of adding new functionality from software view. Last, the 4th section is reserved for conclusions and possible future work.

1. Hybrid Petri Nets

The theory of PNs has its origin in C.A. Petri's dissertation "Communication with Automata", submitted in 1962. PNs are used as describing formalism in a wide range of application fields. They offer formal graphical description possibilities for modeling systems consisting of concurrent processes. PNs extend the automata theory by aspects like concurrency and synchronization.

In the past, there were applications of HPNs described in many cases but essentially, they were concentrated on the fields of process control and automation. In the following we demonstrate the possibilities of using HPNs to model wireless sensor networks infrastructure, which is an embedded hybrid systems. The used PN class of Hybrid Dynamic Nets (HDN) and its object oriented extension is described in [10] and [11]. This class is derived from the above-mentioned approach of David and Alla [12] and defines the firing speed as function of the marking from the continuous net places.

Continuous PNs are particularly suitable for modeling flows: liquid flow, data flows (suitable for WSN), continuous production of a machine. However, a flow may be suddenly interrupted: closing a valve or machine breakdown for example. This is equivalent to suddenly having another continuous PN. This situation can be modeled by a HPN containing continuous places and transitions (**C-places and C-transitions**) and discrete places and transitions (**D-places and D-transitions**). In addition, in a HPN, a discrete marking may be converted into a continuous marking and vice-versa. The marking of a C-place is represented by a real number, whose unit is called a mark, and the marking of a D-place is represented by dots, called *tokens* (or *marks* when a common word is useful).

In Figure 1 is represented an example of HPN, where transition **t1** is enabled only if there is at least one token in **p1**.

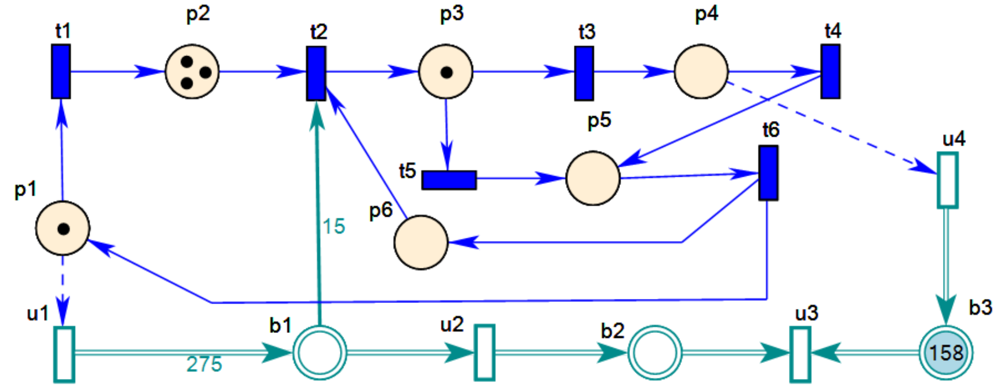

Figure 1 Example of Hybrid petri net.

Discrete part			**Continuous part**		
D-Place Tokens	Immediate transition Timed transition	Normal arc Test arc	**C**-Place 158 Fluid	Continuous transition	Fluid arc Normal arc Test arc

Same convention is used in the Rewriting Hybrid Petri Net tool developed by UTM under the guidance of Dr. Guțuleac Emilian.

The formal definition described in continuation is same as in [12] which is considered to be one of wide used formalization of Petri Nets.

Formal Definition [12]

A marked **autonomous hybrid Petri net** is a sextuple $R = \langle P, T, Pre, Post, m_0, h \rangle$ fulfilling the following conditions:

$P = \{P_1, P_2, \dots, P_n\}$ is a finite, not empty, set of places;

$T = \{T_1, T_2, \dots, T_m\}$ is a finite, not empty, set of transitions;

$P \cap T = \emptyset$, i.e. the sets P and T are disjointed;

$h: P \cup T \rightarrow \{D, C\}$, called "hybrid function", indicates for every node whether it is a discrete node (sets P^D and T^D) or a continuous one (sets P^C and T^C);

$Pre: P \times T \rightarrow Q_+$ or N is the input incidence application;

$Post: P \times T \rightarrow Q_+$ or N is the output incidence application;

$m_0: P \rightarrow R_+$ or N is the initial marking

In the definitions of $Pre, Post,$ and m_0, N corresponds to the case where $P_i \in P^D$, and Q_+ or R_+ corresponds to $P_i \in P^C$. Pre and $Post$ functions must meet the following criterion: if P_i and T_j are such that $P_i \in P^D$ and $T_i \in T^D$, then $Pre(P_i, T_j) = Post(P_i, T_j)$ must be verified.

As for a discrete or a continuous PN, the structure (implying the incidence matrix) is defined by the quadruple $\langle P, T, Pre, Post \rangle$. An *unmarked hybrid PN* is a 5-tuple $Q = \langle P, T, Pre, Post, h \rangle$, i.e., in addition to the structure, the nature of each node (discrete or continuous) is specified in *Q*.

The last condition states that an arc must join a C-transition to a D-place as soon as a reciprocal arc exists. This ensures marking of D-places to be an integer whatever evolution occurs.

Definition 2.

***A* discrete transition** in a hybrid PN is **enabled** if each place P_i in $^*T_j$ meets the condition (as for a discrete PN):

$m(P_i) = Pre(P_i, T_j)$.

Definition 3.

A **continuous transition** in a hybrid PN is **enabled** if each place P_i in $^*T_j$ meets the condition (as for a discrete PN):

$$\begin{cases} m(P_i) \geq Pre(P_i, T_j), & \text{if } P_i \text{ is a D-place} \\ m(P_i) > 0, & \text{if } P_i \text{ is a C-place} \end{cases}$$

2. PIPE modelling software

Models are drawn on a canvas using components from a drawing toolbar including places, transitions, arcs and tokens Figure 2. Nets of arbitrary complexity can be drawn and annotated with additional user information. Besides basic model design functionality, the designer interface provides features such as zoom, export, tabbed editing and animation. The animation mode is particularly useful for aiding users in the intuitive verification of the behaviour of their models. PIPE uses the Petri Net Markup Language (PNML) [13] as its file format, which permits interoperability with other Petri net tools including P3 [14], WoPeD [15] and Woflan [16] and the models are saved into XML files [9].

Central to the architecture of PIPE is its support for modules, which allow its functionality to be extended at runtime with user-implemented code. PIPE comes equipped with a number of specialized analysis modules that perform structural and performance-related analyses on PN models. A panel to the left of the canvas enables users to access these modules.

In Figure 3.1 it can be seen the famous problem of “dinning philosophers” designed in PIPE tool using Petri Net modeling language. This tool permits to simulate and to evaluate the model.

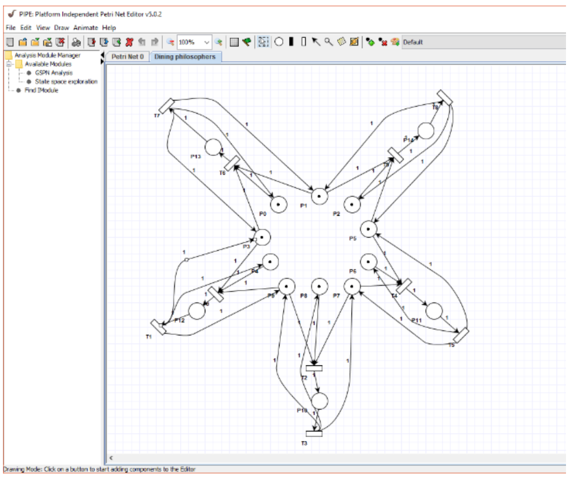

Figure 2. PIPE design interface.

The project is maintained in programming language Java 8, and has a wide community of users. The source code of the project are present on GitHub versioning server [17]. For project building automation is used Maven [18] which permits to add easily a new module (in our case – the proposed plugin).

Conclusion

The feature of modeling HPNs, brings PIPE simulation and evaluation tool to a larger space of problems. One of them are flows and WSN which we propose to evaluate. We are currently implementing and testing this plugin. Future work on this subject can be, graphical representation of analysis, in form of graphs.

References

1. Gupta, Dr. Deepti. “Wireless Sensor Networks ‘Future trends and Latest Research Challenges’.” ISSN: 2278-8735, pp. 41-46, (2015).
2. Yang, Yuanyuan & Wang, Cong & Li, Ji. Wireless Rechargeable Sensor Networks - Current Status and Future Trends. Journal of Communications. 10. 696-706. 10.12720/jcm.10.9.696-706, (2015).
3. Guţuleac E., Zaporojan S., Țurcanu I., Furtuna A. „Analiza Performanţelor Sistemelor de Calcul prin Reţele Petri Hibride Stocastice Aproximativ Agregate”, ISBN 978-9975-45-540-4, pp. 271-275, (2018).
4. A.T. Belmansour, M. Nourelfath, “An aggregation method for performance evaluation of a tandem homogenous production line with machines having multiple failure modes,” Reliability Eng.and System Safety, Vol 95, No 11, pp. 1193-1201, (2010).
5. R. Bierbooms, I. Adan, M. Vuuren, “Approximate performance analysis of production lines with continuous material flows and finite buffers,” Stochastic Models, 29, pp. 1–30, (2013).
6. E. Guţuleac, “Descriptive compositional HSPN modeling of computer systems,” Annals of the University of Craiova, România, Series: Atomation, Computers, Electronics and Mechatronics, Vol. 3(30), No.2, pp. 82-87, (2006).
7. E. Guţuleac, Evaluarea performanţelor sistemelor de calcul prin reţele Petri stochastice. Ed. „Tehnica-Info”, Chişinău, - 276 p, (2004)
8. Hummel, Thorsten & Fengler, Wolfgang & Adamski, Marian & Karatkevich, Andrei & Wegrzyn, Marek. Design of Embedded Control Systems Using Hybrid Petri Nets. 10.1007/0-387-28327-7_12, (2005).
9. J. Dingle, Nicholas & J. Knottenbelt, William & Suto, Tamas. PIPE2: A tool for the performance evaluation of generalised stochastic Petri Nets. SIGMETRICS Performance Evaluation Review. 36. 34-39, (2009).
10. Drath, R.: Modeling Hybrid Systems Based on Modified Petri Nets. PhD Thesis, TU Ilmenau, (1999)
11. Drath, R.: Hybrid Object Nets: An Object-oriented Concept for Modeling Complex Hybrid Systems. In: Hybrid Dynamical Systems. Third International Conference on Automation of Mixed Processes, ADPM'98, Reims, (1998).
12. Alla, H., David, R., Le Bail, J.: Hybrid Petri nets. Proceedings of the European Control Conference, Grenoble, (1991)
13. The Petri Net Markup Language – http://www2.informatik.hu-berlin.de/top/pnml/.
14. D. Gasevic and V. Devedzic, "Software support for teaching Petri nets: P3," Proceedings 3rd IEEE International Conference on Advanced Technologies, Athens, Greece, pp. 300-301, doi: 10.1109/ICALT.2003.1215093, (2003).
15. WoPeD: Workflow Petri Net. Designer – http://www.woped.org/.
16. H. Verbeek and W. van der Aalst, “Woflan 2.0: A Petri-net-based workflow diagnosis tool,” in Proc. 21st International Conference on Application and Theory of Petri Nets (ICATPN’00), vol. LNCS 1825. Springer, pp. 475–484, (2000).
17. PIPE source code - https://github.com/sarahtattersall/PIPE .
18. Apache Maven automation build tool - https://maven.apache.org/ .