A Review of Petri Net Tools and Recommendations

Vinod B. Kumbhar[1](✉) and Mahesh S. Chavan[2]

[1] Shivaji University, Kolhapur, India
vinodkumbhar2012@gmail.com
[2] KIT's College of Engineering (Auto.), Kolhapur, India
chavan.mahesh@kitcoek.in

Abstract. Petri net a scientific modelling language that is used to visually represent a system. It's a commanding language that may be utilized to express concurrent system processes. Users may graphically inspect the entire system and alter it with a Petri net software tool. System's performance may also be examined using a Petri net software tool, produce code, simulate the system, and test the models. This paper will cover the eighteen most recent Petri net tools and offer the readers a description of what a Petri net tool is and describe its key purposes. The reader may learn about the eighteen most recent Petri net tools available from this essay. To this day, several Petri net tools may be downloaded from the internet. This survey article compares 18 Petri net tools from various perspectives. This is significant since consumers won't have to read individual Petri net tool evaluations online. Readers may select best suggested Petri net software tools to employ depending upon their OS and the sorts of Petri net tools to be studied simply by looking at the discussion offered. This survey paper's main goal is to make Petri net tool recommendations according to the types of Petri nets and the operating system that will be examined.

Keywords: Petri net · Petri net Tools · Petri net simulation software

1 Introduction

Carl Adam Petri invented the Petri net in 1962 [1]. Petri created a series of modules, each of which communicated with its two neighbors and contained a single data element [2]. Any control process that can be visually described, such as flow charts that requires a way to express parallel or concurrent operations may be formally represented using a Petri net [3]. Many Petri net software tools have been created for this reason because Petri nets may be used to graphically describe most systems. Users may accurately model their systems and evaluate their performance using Petri net software tools.

The Petri net open source software tools may also be used as a graphic editor and a code generator by users. The system may be simulated and its model checked using several Petri net tools. There are numerous sorts of Petri net tools available right now for a variety of settings and objectives.

S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 710–721, 2023.
https://doi.org/10.2991/978-94-6463-136-4_61

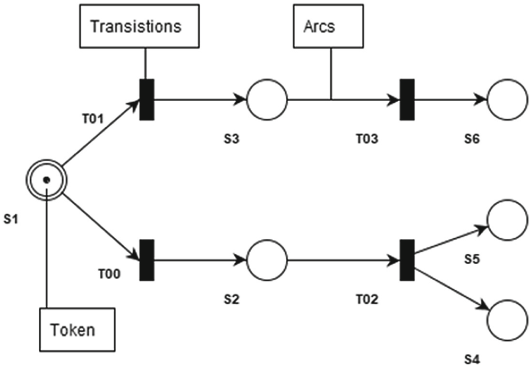

Fig. 1. Sample Petri net

This survey study compares 18 distinct kinds of Petri net software tools from various angles. At the conclusion of the study, recommendations for various Petri net software tools for users of various OS and based on the types of Petri net the researcher desired to analysis is drawn.

2 Preliminaries

Petri net is a formal modelling technique. A solid mathematical foundation and a graphical representational language [4]. It is graphical tool shown, in which, it acts as a tool for visual appearance to simulate the system's operation. Based on a scientific model since it reflects the equations, foundation algorithms, and algebraic equations in the system. In order to simulate control flow in a system, petri dishes are used. These dishes are also suitable for modelling synchronization and concurrency in spread-out systems. Three different component types make up a Petri net: arcs, transitions, and locations (shown as circles) (arrows). Locations stand in for various system states. Transitions are things that happen or do things that alter a situation. A places are connected to transitions via an arc, and vice versa. The token is another component of a Petri net. A change in status is signaled by the moving of a token from one location to another. Token movement is often referred to as fire.

6 places (states S1-S6), 4 transitions (changes in state T0-T3), and nine connecting arcs make up an example Petri net in Fig. 1. For now, token added at S1. Token transition to the following state. The direction the token is travelling is indicated by the arc's arrowhead. Through T1, a token may be launched into s2. The token has left s1 and is currently in s2. The firing continues as the token is shot via T3 to positions s4 and s5. Depending on the quantity of concurrent nodes, this is also mentioned to as a concurrency relationship where Two tokens can be obtained from a single token. Following this procedure, a token is present in the s4 and s5 states, respectively. There are now no tokens in the states of s1 or s2.

Another illustration is the ability of the token in state s1 to be fired into state s3 through T2. As a consequence, s1 is empty and s3 has a token. T4 then launches the token from s3 into s6 to complete the operation from s3 to s6.

Using a 1992 study by L. Bernardinello and F. De Cindio as her foundation, Monika Trompedeller suggested a taxonomy of Petri nets in 1995 [5]. The categories haven't been revised since then, but they're still helpful for receiving a brief rundown of the key distinctions across numerous Petri net types. Petri nets can be divided into three levels; level 1 is made up of places (states) which represent Boolean values; for e.g., a place is designated by a maximum of one unorganized character.

3 Survey

This report provides a comprehensive evaluation of Petri net software packages. In all 18 latest Petri net tools have been discussed here based on various parameters.

3.1 CPN Tool

CPN Tools is widely used for a variety of coloured Petri net operations, including manipulation, simulation, and analysis. Advanced interaction techniques are the foundation of GUI for CPN Tools, which is developed in collaboration with top HCI specialists [6]. Feedback facilities show the dependencies between net parts and give relevant error notifications. While a net is being built, the tool has incremental syntax checking and code-generating functions. Both timed and untimed nets may be handled well by a quick simulator. A typical state space report includes details like boundedness qualities and liveness properties and may produce and analyze whole and partial state spaces [7].

3.2 ePNK

Based on the PNML transfer format, ePNK provides platform for Petri net tools. Offering generic Petri net types that are easy to plug in and a standard GMF editor that can sometimes be utilized to graphically edit nets of any plugged-in type are the main objectives of the project. You can connect other features. The ePNK is implemented using the Eclipse platform, which works with any types of hardware. All Petri net categories distinct by ISO/IEC 15909-2 and PNML are supported by the most recent release of ePNK (version 0.9). The version number indicates that while the ePNK is now stabilizing, some polish is still required [8].

3.3 GPenSIM

GPenSIM is a Perti net software available on MATLAB platform. GPenSIM aims to be an easy-to-use and extensible tool for modelling, simulation, evaluation of performance, control of real time discrete event systems, and for merging MATLAB toolboxes with Petri net modelling like Fuzzy Logic [6, 9].

3.4 HiQON-Tool

The HiQPN Petri net Tool is used to identify and analyze Hierarchically Combined Queueing Petri nets (HQPNs) is mutual Petri net and queuing network modelling formalism. The hierarchical model definition supported by HiQPN-Tool works in conjunction with qualitative and quantitative (performance) analysis and greatly benefits from it [10].

3.5 IOPT-Tools

The creation of digital embedded controllers is made easier by a cloud-based toolchain called IOPT-Tools, which provides a full set of Petri net tools with an online interface. Place-Transition Input-Output Petri nets (IOPT-nets) employ an underlying Petri net class that takes into consideration a maximum step and single server semantics, allows input/output signals events, and makes use of cycle accurate execution. Among the available tools are simulator and model-checking subsystem comprised of a state-space generator, a state-space visualization system, and query system, along with tools for automatic code generation which generate software with C code/VHDL hardware descriptions suitable for deployment on implementation platforms [6, 11].

3.6 WoPeD

For managing, editing, simulating, and evaluating workflow nets using "van der Aalst" notation, WoPeD is a simple, portable programme. Additionally, supported are "classical" P/T nets. WoPeD's primary goals are educational and publishing applications. A complete graphical editor, a visual token game simulator, and interfaces to numerous additional tools by means of the PNML standard are all included in WoPeD. Several qualitative and quantitative analysis techniques are offered for workflow networks. WoPeD is platform-neutral; numerous platforms have been successfully tested with the executables. Installer packages are accessible for systems running Windows, Linux, and MacOS X [12].

3.7 Wolfgang

With the help of Wolfgang, users can quickly develop and amend Petri nets and verify in contrast to both generic and workflow-specific net attributes. With regard to Place-Transition Nets (P/T-Nets) and Colored Petri Nets (CPN) as they are specified in SEPIA library, Wolfgang offers complete PNML conform editing capability. Petri nets may be swiftly created from scratch, graphically edited in a variety of ways, stored, and even exported as PDFs thanks to its user-friendly interface. Make your desired Petri net easily with Wolfgang [13].

3.8 ITS Tools

Under discrete time assumptions, ITS tools enable HLPN and P/T nets that are PNML compatible as well as Time PN. The editing and hierarchical composition of nets is made possible using an eclipse-based front-end. A robust symbolic checker covering safety, CTL, and LTL features is included with the ITS tools [14].

3.9 TAPAAL

The computer science department at AALborg University in Denmark created TAPAAL, it is a modelling, simulating, and verifying Timed-Arc Petri nets that is accessible on Windows, Linux, and Mac OS X platforms.

The TAPAAL programme provides a visual editor to create TAPN models, a simulator for testing out planned nets, and a confirmation environment which provides automatic responses to logical questions written in a subset of CTL logic [6]. Additionally, it enables the user to determine if a particular net is k-bounded for a specific value of k. Three open source verification engines are now included in the most recent version of TAPAAL and are supplied with it. Two-player games can also be modelled, both with and without time components. The user can choose to utilize the UPPAAL verification engine and automatically convert TAPAAL models into UPPAAL [15].

3.10 SYROCO

CoOperative Objects (COO) classes are efficiently converted into C++ classes using the CoOperative Objects compiler. It is appropriate for both the implementation of concurrent applications and the modelling of distributed systems. The fact that COOs support dynamic instantiation, synchronous and asynchronous communications, inheritance, temporization, and data processing makes them (Very) High Level Petri Nets [6, 16].

3.11 SNAKES

SNAKES is acronym for the Net Algebra Kit for Editors and Simulators. It is a Python package that offers everything required to define and run a variety of Petri net types, especially those of the Petri Box Calculus and M-nets families. Its primary goal is to serve as a universal Petri net library, which can design and execute the majority of Petri net models and give researchers a tool to swiftly prototype their novel concepts. Additionally, SNAKES must to be capable of offering the data model to editors or simulators [17].

3.12 SimHPN

SimHPN provides a set of tools for simulating, analyzing, and synthesizing discrete event systems that are described by continuous and hybrid Petri nets. It is integrated within MATLAB environment, which has the significant benefit of enabling the development of robust algebraic, statistical, and graphical instruments that take advantage of the high calibre MATLAB routines [18]. The software package's key characteristics are:

- Continuous Petri net simulation with the aforementioned firing semantics Product semantics, finite servers, and infinite servers, Add-ons for several graphical Petri net editors,
- possibilities for various visualizations, computing the throughput limits, P-T-semi flow computation.
- optimum sensor positioning, Perfect steady-state.

3.13 WINSIM

WINSIM models all types of information systems using a class of advanced Petri nets with time, data processing, attributed tokens, and control competencies. With internal functions for generating random integers and random variates, it supports Modeling Control Language (MCL) and its accompanying Model Description Language (MDL) (MCL). The creative and executive subsystems make up the overall system. The MDL compiler is used by the creative subsystem to prepare and change the source texts for MDL models, translate those texts into executable models, and integrate those models into the system [19].

3.14 WebSPN

It is a modelling tool to study non-Markovian stochastic Petri nets, unlike some of the current tools' shortcomings. This tool supports the analysis of a larger class of Petri net models with simultaneously enabled widely scattered transitions in the prd, prs, and pri variables. WebSPN offers a discrete-time estimate of the model's stochastic behavior. WebSPN makes considerable use of Java technology, so any Internet-connected node with a Java-enabled web browser can readily access it. The solution engine may run across a number of workstations thanks to the kernel's message-passing MPI-2-based parallel approach [20].

3.15 COSMOS

Cosmos supports a variety of file formats in addition to GrML, including.pnml, the ISO standard for Petri Net files,.gspn, a text-based Petri Net file format, and.andl, the file format for the model checker Marcie. Prism is also partially supported. The.PNPRO file format, which is utilized by the GreatSPN Editor to provide a straightforward way to see and modify Petri Nets, is also provided in the files that follow [6, 21].

3.16 Platform Independent Petri Net Editor 2

Quickly, effectively, and efficiently create Petri Nets. There are now 6 analysis modules, including simulation analysis, state-space analysis (deadlock analysis), and invariant investigation. An important design element is an extendable element interface that makes it simple to write additional analysis modules. Completely complies with PNML, despite the file format's ability to be extended via XSLT [22].

3.17 PNetLab

PNetLab is Simulation and analytical tools that can be applied in both scenarios and aid in supervisor training are in low supply. The University of Salerno's Automatic Control Group created the simulation and analysis programme PNetLab. PN/CPN/CMHPN models may be created using a graphical user interface, and supervisor can either be PN/CPN or a typical C/C++ programme that performs logical predicates [23].

3.18 SIPN-Editor

An example of a software tool for editing, visualizing, and converting HSIPN is the SIPN-Editor (SIPNs). The primary element of created SIPN tool is a graphical editor that makes it simple to alter SIPNs by direct manipulation. The tool's major usage is as a programming tool for PLC programmers. As a programming tool prototype, the SIPN tool converts produced SIPNs into corresponding IL programmes that implement the SIPNs on PLCs. The SIPN tool also deals with a code generator option for making verifiable SMV code [24].

4 Conclusions and Discussions

Table 1 and Table 2 displays the findings of the survey. Five criteria—Petri net support, component, performance evaluation, environments, and if it is free—are used to underpin the analysis. Petri net tools evaluated in Sect. 3 are compared in the table above based on five key factors. The Petri net supports the first set of requirements. In this category, the supported kinds of Petri nets for each Petri net tool are contrasted.

With certain supportive high-level Petri net (such as GpenSIM and COSMOS, the mainstream of the tools provides Place/Transition Petri net, high level, timed petri net, and stocostic). But because it offers stocostic net, GpenSIM shines out in this area (a combination of Stocostic and Petri net). Only SIPN editor supports signal interpreted petri net.

The components of each tool serve as the second comparative criterion. The majority of the programs include a visual editor and a quick Petri net simulation. Offering a visual editor and quick Petri net simulation might be useful teaching resources. To aid in the learning of Petri net, users will be able to alter and simulate numerous Petri nets. Compared to the other technologies.

The analysis of a Petri net is the next comparative category. Simple performance analysis is offered by several of the tools reviewed, whereas structural analysis is offered by tools like GPenSIM, Wolfgang and Netlab. Additionally, capable of performing advance performance analysis are GpenSIM, HiQPN, TAPAAL, SimHPN and WebSPN. The environment is the next factor used to evaluate different Petri net tools. With six environments a piece, WINSIM, SimHPN and SYROCO have the most environments supported for their tools. The cost of Petri net tools serves as the last comparative criteria. All Petri net tools are either one free to use or may be downloaded for free for educational purposes.

In this part, Table 3, which comprises the users' operating systems and the categories of Petri net to be studied, is offered to suggest a Petri net tool for a few cases.

Table 2 provides an overview of the suggested Petri net tools for various settings and categories of analyzing a petri net. In general, users of the utility are suggested for usage with MAC OS X.

PIPE is also advised for Windows users as it is compatible for majority of Petri net types. However, users must use the tools GpenSIM and COSMOS, respectively, to analyze stocostic Petri nets and timed Petri nets. There isn't a single programme that can assess the majority of the many Petri net kinds specifically for Linux users.

Table 1. Findings of the Survey (Part A)

	Petri Net Tool	CPN Tool	ePNK	GPenSIM	HiQPN-Tool	WoPeD	Wolfgang	ITS Tool	TAPAAL	SYRACOS	SNAKES	SimHPN	WINSIM	WebSPN	COSMOS	PNetLab	SIPN-Editor	IOPT-Tools	PIPE
Components	Graphical Editor	√	√		√	√	√	√	√	√				√		√	√	√	√
	State Spaces	√	√	√	√		√	√			√			√				√	
	Condensed State Space							√			√			√					
	Interchange File Format	√			√	√	√		√	√	√				√	√		√	√
	Code Generation									√							√	√	√
	Petri Net Generator																		√
	Place invariants			√	√											√			
	Transition invariants			√	√											√			√
	Token game Animation	√				√	√		√					√		√		√	
	Model Checking		√																√
	Fast Simulation	√								√	√				√			√	√
Petri net Analysis	Invariant based analysis																		
	Advance Performance analysis			√	√				√			√		√					√
	Structural analysis			√			√									√			√
	Simple performance analysis	√		√						√						√			√
	Reachability graph analysis									√									

Table 2. Findings of the Survey (Part B)

	Petri Net Tool	CPN Tool	ePNK	GPenSIM	HiQPN-Tool	WoPeD	Wolfgang	ITS Tool	TAPAAL	SYROCO	SNAKES	SimHPN	WINSIM	WebSPN	COSMOS	PNetLab	SIPN-Editor	IOPT-Tools	PIPE
Petri Net type Supported	Signal Interpreted Petri Nets																√		
	Place transition nets		√			√	√	√				√				√		√	√
	Stocostic			√										√	√				
	Time Petri nets	√						√		√						√			√
	High Level Petri Nets	√	√	√			√	√		√		√	√	√		√			√
Environment	Java		√			√	√		√				√	√			√		
	Macintosh					√	√		√		√	√			√			√	√
	Linu	√					√		√		√	√	√	√	√			√	
	MS DOS								√	√	√	√							
	HP, HP-U								√		√	√							
	Silicon Graphics, IRRI								√		√	√							
	Windows	√		√		√	√		√	√	√	√	√	√		√		√	
	Sun				√				√		√	√							√
	Uni																		
Free		√		√	√	√	√	√	√		√	√	√	√	√	√	√	√	√

Users are advised to utilize the PEP tool for High-Level Petri nets, Petri nets with time, and Place/Transition Petri nets. Users are advised to utilize GreatSPN or COSMOS for stochastic petri net.

For analysis of a signal interpreted Petri net, SIPN editor is advised. Overall, among the 18 Petri net tools reviewed, the PIPE tool provides the most components and analysis kinds. PIPE, is also user-friendly owing to the lack of a supportive atmosphere. To use the PEP tool, users must have a Linux/Sun operating system.

Table 3. Overview of the suggested Petri net tools

Type of Operating System	Petri Net types	Suggested Tool
MAC OS X	Object Oriented	Not Supported
	High Level	Wolfgang
	Stocostic	COSMOS
	Timed	PIPE
	Place Transition Nets	COSMOS
	Transfer	Not Supported
Windows OS	High Level	PIPE
	Object Oriented	PIPE
	Stocostic	COSMOS
	Time	PIPE
	Place transition Nets	PIPE
	Transfer	Not Supported
Linux OS	High Level	PIPE
	Object Oriented	Not Supported
	Stocostic	COSMOS
	Time	PIPE
	Place Transition Nets	PIPE

Furthermore, PIPE is the tool with the highest module and analysis kinds and covers the majority of the key environments (Windows, MAC OS X, and Linux). According to Table 2, every operating system contains two different types of Petri nets that may be examined by PIPE. This deduction leads us to the conclusion that PIPE is the most suitable tool since it supports the majority of operating systems and has a sufficient number of capabilities.

References

1. Petri, C. A. Kommunikation mit automaten. (1962).
2. Brauer, W., and Reisig, W. Carl Adam Petri and “Petri nets”. Fundamental Concepts in Computer Science, 3: 129-139 (2009).
3. Murata, T. “Petri nets: Properties, analysis and applications,” Proceedings of the IEEE, 77(4): 541- 580 (1989).
4. Ameedeen, M. A. A model driven approach to analysis and synthesis of sequence diagrams (Doctoral dissertation, University of Birmingham) (2012).
5. Trompedeller, M. A Classification of Petri Nets. (1995).
6. R. Zierke, “Fachbereich Informatik,” www.inf.uni-hamburg.de. https://www.informatik.uni-hamburg.de (accessed Nov. 10, 2022).

7. Westergaard, M., Kristensen, L.M. The Access/CPN Framework: A Tool for Interacting with the CPN Tools Simulator. In: Franceschinis, G., Wolf, K. (eds) Applications and Theory of Petri Nets. PETRI NETS 2009. Lecture Notes in Computer Science, vol 5606. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02424-5_19 (2009).
8. Ekkart Kindler: The ePNK: A generic PNML tool - Users' and Developers' Guide for Version 1.0.0. IMM-Technical Report-2012–14, DTU Informatics, Kgs. Lyngby, Denmark, December 2012 (revised and extended version of IMM-Technical Report-2011–03). (2012).
9. R. Davidrajuh, "GPenSIM: A New Petri Net Simulator", in Petri Nets Applications. London, United Kingdom: IntechOpen, [Online]. Available: https://www.intechopen.com/chapters/9199 doi: https://doi.org/10.5772/7499 (2010).
10. Bause, Falko & Buchholz, Peter & Kemper, Peter. QPN-Tool for the Specification and Analysis of Hierarchically Combined Queueing Petri Nets. Quantitative Evaluation of Computing and Communication Systems. 977. https://doi.org/10.1007/BFb0024318. (1995).
11. Pereira, Fernando & Moutinho, Filipe & Costa, Anikó & Barros, João-Paulo & Campos-Rebelo, Rogerio & Gomes, Luis. IOPT-Tools – From Executable Models to Automatic Code Generation for Embedded Controllers Development. https://doi.org/10.1007/978-3-031-06653-5_7. (2022).
12. Freytag, Thomas & Sänger, Martin. WoPeD - An Educational Tool for Workflow Nets. CEUR Workshop Proceedings. 1295. 31. (2014).
13. "Wolfgang," Uni-freiburg.de. [Online]. Available: http://doku.telematik.uni-freiburg.de/wolfgang. [Accessed: 13-Nov-2022].
14. Yann Thierry-Mieg. General overview of ITS-tools Symbolic Model-Checking Using ITS-Tools in Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). (2015)
15. Byg, J., Jørgensen, K.Y., Srba, J. TAPAAL: Editor, Simulator and Verifier of Timed-Arc Petri Nets. In: Liu, Z., Ravn, A.P. (eds) Automated Technology for Verification and Analysis. ATVA. Lecture Notes in Computer Science, vol. 5799. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04761-9_7 (2009).
16. Mikolajczak, B. & Sefranek, C.A. Integrating object oriented design with concurrency using Petri nets-A case study of a banking system and the Syroco-Macao environment. 3. 1559 - 1564 vol.3. https://doi.org/10.1109/ICSMC.2001.973506. (2001).
17. Pommereau, Franck. SNAKES: A flexible high-level Petri nets library. 9115. https://doi.org/10.1007/978-3-319-19488-2_13. (2015).
18. Júlvez, Jorge & Mahulea, Cristian & Vazquez, Carlos. SimHPN: A MATLAB toolbox for simulation, analysis and design with hybrid Petri nets. Nonlinear Analysis: Hybrid Systems. 6. 806–817. https://doi.org/10.1016/j.nahs.2011.10.001. (2012).
19. Thong, Weng Jie & Ameedeen, Mohamed. A Survey of Petri Net Tools. Lecture Notes in Electrical Engineering. 315. 537-551. https://doi.org/10.1007/978-3-319-07674-4_51. (2015).
20. Longo, Francesco & Scarpa, Marco & Puliafito, Antonio. WebSPN: A Flexible Tool for the Analysis of Non-Markovian Stochastic Petri Nets. https://doi.org/10.1007/978-3-319-30599-8_10. (2016).
21. P. Ballarini, B. Barbot, M. Duflot, S. Haddad and N. Pekergin. HASL: A New Approach for Performance Evaluation and Model Checking from Concepts to Experimentation. Performance Evaluation 90, pages 53-77, (2015).
22. Liu, Su & Zeng, Reng & He, Xudong. PIPE + - A modeling tool for high level Petri nets. 115–121. (2011).

23. Basile, Francesco & Carbone, Ciro & Chiacchio, Pasquale. Simulation and analysis of discrete-event control systems based on Petri nets using PNetLab. Control Engineering Practice - CONTROL ENG PRACTICE. 15. 241–259. https://doi.org/10.1016/j.conengprac.2006.07.006. (2007).
24. Klein, Stéphane & Frey, Georg & Minas, Mark. PLC Programming with Signal Interpreted Petri Nets. 2679. 440-449. https://doi.org/10.1007/3-540-44919-1_27. (2003).