A SURVEY OF PETRI NET TOOLS

Thong Weng Jie and Mohamed Ariff bin Ameedeen
Fakulti Komputer Sistem and Kejuruteraan Perisian, Universiti Malaysia Pahang, Kuantan, Pahang, Malaysia
E-Mail: 1briantwj@gmail.com

ABSTRACT

Petri net is a mathematical modeling language used to describe a system graphically. It is a strong language that can be used to represent parallel or concurrent activities in a system. With a Petri net tool, users can view the overall system graphically and edit it with the editor. A Petri net tool can also be used to analyze the performance of the system, generate code, simulate the system and perform model checking on it. A review on twenty Petri net tools in this paper will give the readers an idea on what is a Petri net tool and the main functions of a Petri net tool. This paper can serve as an introduction of twenty Petri net tools to the reader. However, to date, there are many Petri net tools available to be downloaded online. This survey paper aims to compare twenty Petri net tools in different aspect. This is important as users will not have to read the reviews of Petri net tools online one by one. Just be having a look at the discussion provided, readers can determined the best recommended Petri net tools to be used based on their operating systems and the types of Petri net tool to be analyzed. The main purpose of this survey paper is to recommend Petri net tools based on the operating system and the types of Petri net to be analyzed.

Keywords: Petri net, Petri net tools.

INTRODUCTION

Petri net were introduced by Carl Adam Petri in 1962 [1]. Petri designed a sequence of modules, with each module containing a single data element and communicating with its two neighbors [2]. Petri net can be applied informally to any system that can be described graphically like flow charts and that needs some means of representing parallel or concurrent activities [3]. Since Petri net can be applied in most system to characterize it graphically, a lot of Petri net tools had been developed for this purpose. Using Petri net tools, users can represent their system in details and analyses the performance of the system. Users can also use the Petri net tools as a graphical editor and code generator. Some Petri net tools can also be used to simulate the system and provide model checking for it. To date, there are many different types of Petri net tools for different environments and purposes. However, there are no published papers on the recommendation of Petri net tools to the users based on the users' operating system and Petri net to be analyzed. This survey paper aims to compare twenty different types of Petri net tools in different aspects. At the end of the paper, a discussion will be drawn to recommend different types of Petri net tools to users with different operating systems and depending on what type of Petri net the users wanted to analyze.

PRELIMINARIES

Petri net can be defined as a formal modeling language that can be represented graphically with a strong mathematical foundation [4]. It is represented graphically in the sense that it serves as a visual communication aid to model the system behavior. It is based on a mathematical foundation in the sense that it represents the equations, algebraic equations and algorithms in the system. Petri net are used to model control flow in a system and is capable of modeling concurrency and synchronization in distributed systems.

A Petri net consists of three types of components: *places* (circles), *transitions* (rectangles) and *arcs* (arrows). Places represent different states of the system. Transitions represent events or actions which cause the change of a state. An arc connects a place with a transition or a transition with a place. Another element in Petri net is the token. The movement of a token from place to place indicates a change of state. The movement of a token is also known as firing.

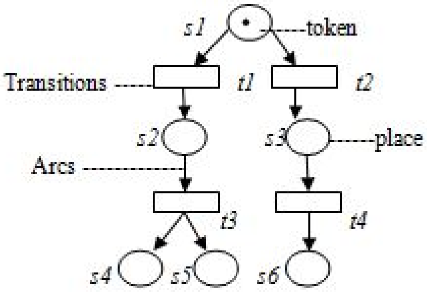


Figure-1. Example of Petri net.

Figure-1 represents an example of a Petri net with six places (states), four transitions (change of states) and nine connecting arcs.

The token is at *s1* for starters. A token travels to the next state via transition. The direction of the token's movement is represented by the arrow head of the arcs. The token can be fired into *s2* via *t1*. The token is now present in *s2* after leaving *s1*. The firing continues with the token bring fired to *s4* and *s5* via *t3*. This is also known as a concurrency or parallel relationship where one token is split into two (depending on the number of concurrent nodes). After this process, both the *s4* and *s5* states contain a token each. The *s1* and *s2* states are now without any tokens.

For another example, the token is *s1* can also be fired into *s*3 state via *t2*. This result in *s3* contains a token and *s1* being empty. The process continues with *t4* firing the token from *s3* into *s6*.

In 1995, Monika Trompedeller proposed a classification of Petri net based on a survey by L. Bernardinello and F. De Cindio from 1992 [5]. The classifications have not been updated since then but still it is useful for getting a quick overview of the main differences between various kinds of Petri net. Petri net can be classified into 3 levels; in level 1, Petri net is characterized by places (states) which can represent Boolean value, for example, a place is marked by at most one unstructured token. Examples of level 1 Petri net are Condition/Event (C/E) Systems, Elementary Net (EN) Systems and 1-safe systems. An example of Petri net tool that supports analysis of level 1 Petri net is the Environment for Action and State based Equivalences (EASE).

For level 2, Petri net is characterized by places (states) which can represent integer values, for example, a place is marked by a number of unstructured tokens. An example of level 2 Petri net is the Place/Transition (P/T) Nets. Petri net tools that support analysis of level 2 Petri net include CPN/AMI, MOBY, POSES, PNS and WINPETRI.

In level 3, Petri net is characterized by places (states) which can represent high-level values, for example, a place is marked by a multi-set of structured tokens. Examples of level 3 Petri net include High-Level Petri net with Abstract Data Types (HL+ADT), Environment Relationship (ER) Nets, and Traditional High-Level Petri net and Well-Formed (Colored) Nets (WN). An example of Petri net tools that support level 3 Petri net analyzing is the Cabernet which is used to analyze real time systems based on Petri net augmented with data, predicates, actions, and temporal information.

Judging by the classifications of Petri net and their complexity, different Petri net analyzing tools can only support different level of Petri net. There is not any best or worst tool in analyzing Petri net. Different categories of Petri net require different Petri net tools to analyze. For example a level 2 P/T Petri net will require tools such as CPN/AMI to analyze it instead of a level 3 Petri net tool such as Cabernet.

SURVER

In this paper, several Petri net tools are compared.

AlPiNA

AlPiNA (Algebraic Petri net Analyzer) is a model checker for Algebraic Petri net created by the SMV (Software Modeling and Verification) Group at the University of Geneva [6]. AlPiNA is fully written in Java and it is available under the terms of the GNU general public license. AlPiNA provides a user-friendly user interface that was built with the latest metamodeling techniques on the eclipse platform. An Algebraic Petri net is categorized as a level 3 High-Level Petri net with Abstract Data Types (HL+ADT). Hence, AlPiNA is a Petri net tool which is capable of analyzing a level 3 Petri net.

Coopn builder

CO-OPN stands for Concurrent Object-Oriented Petri net. CoopnBuilder is an environment composed of a set of tools destinated to the support of concurrent software development based on the CO-OPN language [7]. CoopnBuilder is also a research project from the SMV Group. CO-OPN is an object-oriented specification language based on synchronized Algebraic Petri net. CO-OPN allows the definition of active concurrent objects, and includes facilities for sub-typing, sub-classing and genericity.

Great SPN

GreatSPN2.0 is a software package for modeling, validation and performance evaluation of a distributed system using Generalized Stochastic Petri net and their colored extension: Stochastic Well-formed Nets [8]. It provides a friendly framework to experiment with timed Petri net based modeling techniques. GreatSPN implements analysis algorithms to allow its use on complex application.

LoLA

LoLA is a low level Petri net analyzer [9]. It is able to verify a broad variety of behavioral properties on a given Petri net. It is based on explicit state space verification, for example, an exhaustive enumeration of reachability state. In the context of service-technology.org, LoLA can be used to verify compatibility and for the validation of formal semantics.

PEP

PEP (Programming Environment based on Petri net) is a comprehensive set of modeling, compilation, simulation and verification components, linked together within a Tcl/Tk-based graphical user interface [10]. PEP's modeling components facilitate the design of parallel systems by parallel program, the compiler then generate Petri net from such models. The PEP tool is considered as an open platform. Further algorithms can be integrated in the user interface.

Snoopy

Snoopy is software tool to design and animate hierarchical graphs, among others Petri net [11]. The tool has been developed and is still under development at the University of Technology in Cottbus, Dep. of Computer Science, “Data Structures and Software Dependability”. The tool is used for the verification of technical systems, typically software-based systems, as well as for the validation of natural systems such as biochemical networks as metabolic, signal transduction etc.

Marcie

Marcie (Model Checking and Reachability analysis done efficiently) is a tool for qualitative and quantitative analysis of a Generalized Stochastic Petri net with extended arcs [12]. The tool consists of four engines with each engine carrying their own unique function. It is possible to use Marcie with a graphical user interface provided by the tool Charlie.

Charlie

Charlie is a software tool used to analyze a level 2 Petri net in particularly a Place/Transition Net [13]. The tool has been developed and is still under development at the University of Technology in Cottbus, Dep. of Computer Science, “Data Structures and Software Dependability”. The main features of the Petri net tool include structural properties analysis, invariant based analysis, reachability graph based analysis, reachability/coverability graph visualization using the JUNG library and plugin support. Charlie is able to read Place/Transition nets which have been created by the tool Snoopy, or P/T nets which are given in the Abstract Petri net Notation and also P/T nets that are given in the INA file format.

JSARP

JSARP (Simulator and Analyzer Petri net in Java) is a Petri net tool that describes and verifies Petri net with the support of a graphical interface [14]. JSARP is developed in Java and employs modern object oriented techniques and design patterns. JSARP is able to edit the Petri net with graphics and also works as a simulator.

MIST

MIST is a safety checker for Petri net and other extensions [15]. It implements several algorithms that solve the coverability problem for monotonic extensions of Petri net. The tool implements several algorithms: forward/backward search of the state using a symbolic data structure (the IST library). The tool also implements abstraction-refinement techniques or efficient traversal techniques to tackle the state explosion problem.

PETRUCHIO

Petruchio is a tool used for computing Petri net translations of dynamic networks modeled in terms of Pi-calculus processes [16]. It provides a mean to further analyze nets. Petruchio is able to simulate GSPN (Generalized Stochastic Petri net) which is a high level Petri net, check coverability for low-level Petri net and application of reduction techniques for low-level Petri net.

PNEditor

PNEditor is an open-source Petri net editor [17]. It offers usual features of a graphical editor for the design of Place/Transition nets. In addition, the tool offers features like saving the net to a file, definition of roles, definition of subnets (nested nets), saving of predefined subnets to files and their reuse as reusable components, replacement of subnets, definition of static places etc. It requires JAVA SE 6+ to operate.

Yasper

Yasper (Yet another Smart Process Editor) is a tool for modeling and simulating stepwise process [18]. It uses extended Petri net as its modeling technique. Yasper offers easy editing, token gameplay and performance analysis with randomized automatic simulation for basic Place/Transition nets with some extension, including case-specific vs. inter-case token flow, reset and inhibitor arcs, decision nodes with parameterized probabilities of alternatives etc.

PAPETRI

PAPETRI is a general and integrated environment for editing and analyzing Petri net [19]. It allows users to work with difference classes of nets. Several analysis tools are available for each of these classes. PAPETRI aims to provide a friendly editing environment and to afford a greater deal of analysis tools for different classes of Petri net.

Xpetri

XPetri is a graphical simulator of Petri net [20]. XPetri is an Xwindows application designed to be portable across UNIX platform. It supports Place/Transition nets with weighted arcs and a strict firing rule. It also allows a minimum and maximum number of tokens to be specified for each place. Xpetri also supports temporary disabling of transitions for a single fire or until re-enabled.

PROD

PROD is a reachability analysis tool for Predicate Transition Nets [21]. The stubborn set method for reduced state space generation has been implemented in PROD. PROD also has a rich query language for inspecting the generated state space. It is available for download and free of charge.

ARP

ARP (Petri net Analyzer) is a software tool for Petri net analysis and simulation developed by Prof. Carlos A. Maziero [22]. The interface is simple and the ARP tool accepts Place/Transition Nets, Timed Nets and Extended Timed Nets. The features of ARP include accessibility analysis, invariant analysis, equivalence analysis, performance evaluation and manual simulation.

JPetriNet

JPetriNet is software that is used to model, analysis conventional Petri net and to simulate Timed Petri net [23]. The project is a Petri net modeling, analysis and simulation tool made in Java Programming Language. The tool is created for educational purpose and also to be used in any other purpose involving concurrent system.

Petri.NET Simulator

Petri.NET Simulator is a tool used for modeling and simulation of Petri net and analysis of their behavior [24]. It can be used to simulate flexible manufacturing systems and also be used for discrete event system. The tool is free for download.

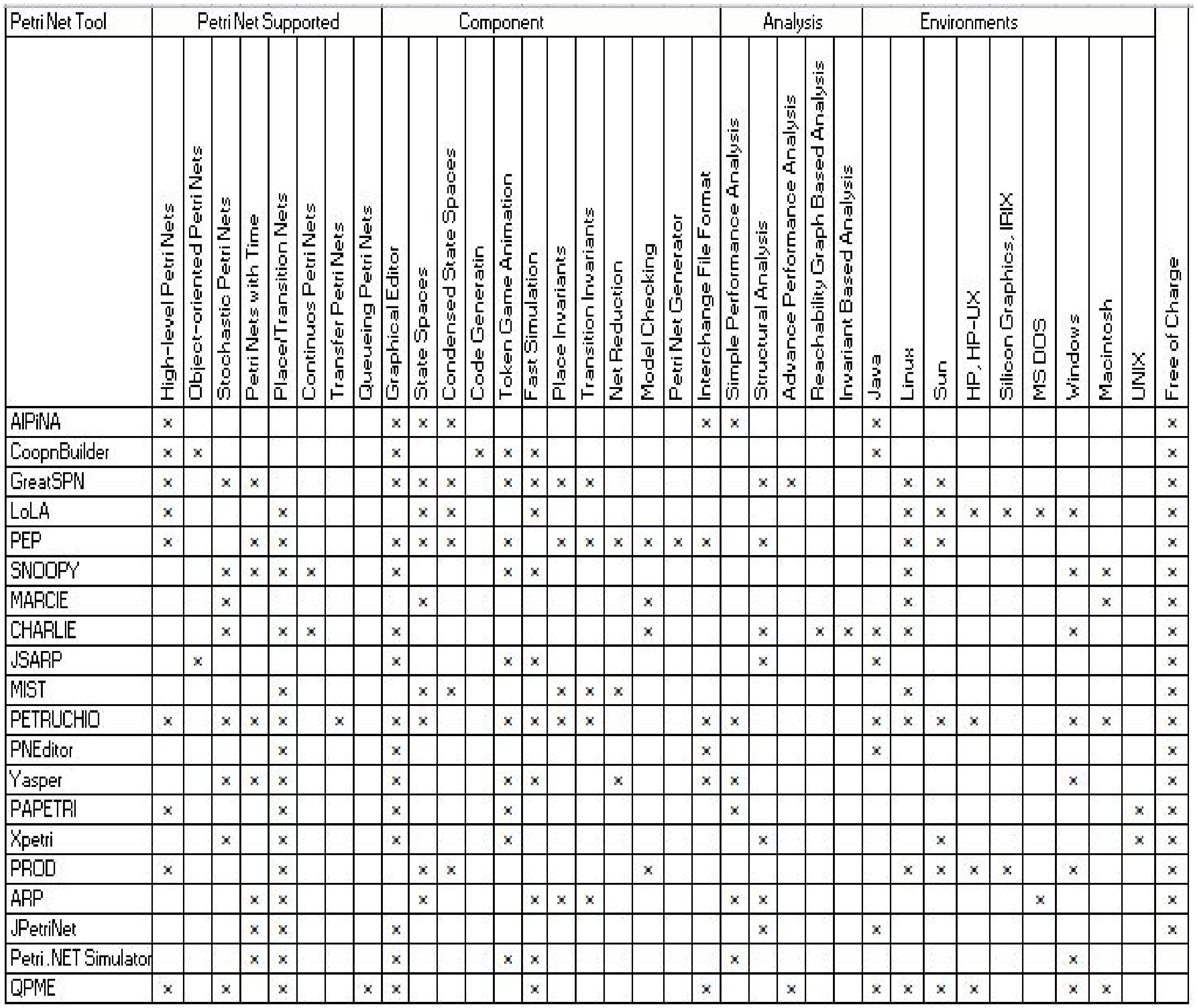
QPME

QPME (Queuing Petri net Modeling Environment) is an open-source tool for stochastic modeling and analysis based on the QPN (Queuing Petri net) modeling formalism [25]. Queuing Petri net is a combination of conventional queuing networks and stochastic Petri net which provides improved expressiveness and thus making it possible to model systems at a higher degree of accuracy. QPME is made of two components, which is QPE (QPN Editor) and SimQPN (Simulator for QPN).

DISCUSSIONS AND CONCLUSIONS

In this section, the survey is tabulated in Table-1. Analysis is done based on five criteria, which are Petri net supported, Component, Analysis, Environments and whether it is Free of Charge.

Table-1. Comparison between Petri Net Tools.



Petri Net Tool	Petri Net Supported								Component												Analysis					Environments									
	High-level Petri Nets	Object-oriented Petri Nets	Stochastic Petri Nets	Petri Nets with Time	Place/Transition Nets	Continuos Petri Nets	Transfer Petri Nets	Queueing Petri Nets	Graphical Editor	State Spaces	Condensed State Spaces	Code Generatin	Token Game Animation	Fast Simulation	Place Invariants	Transition Invariants	Net Reduction	Model Checking	Petri Net Generator	Interchange File Format	Simple Performance Analysis	Structural Analysis	Advance Performance Analysis	Reachability Graph Based Analysis	Invariant Based Analysis	Java	Linux	Sun	HP, HP-UX	Silicon Graphics, IRIX	MS DOS	Windows	Macintosh	UNIX	Free of Charge
AlPiNA	x								x	x	x									x	x					x									x
CoopnBuilder	x	x							x			x	x	x												x									x
GreatSPN	x		x	x					x	x	x		x	x	x	x						x	x				x	x							x
LoLA	x				x					x	x			x													x	x	x	x	x	x			x
PEP	x			x	x				x	x	x		x		x	x	x	x	x	x		x					x	x							x
SNOOPY			x	x	x	x			x				x	x													x					x	x		x
MARCIE			x							x								x									x						x		x
CHARLIE			x		x	x			x									x				x		x	x	x	x					x			x
JSARP		x							x				x	x								x				x									x
MIST					x					x	x				x	x	x										x								x
PETRUCHIO	x		x	x	x		x		x	x			x	x	x	x				x	x					x	x	x	x			x	x		x
PNEditor					x				x											x						x									x
Yasper			x	x	x				x				x	x			x			x	x											x			x
PAPETRI	x				x				x				x								x													x	x
Xpetri			x		x				x				x									x						x						x	x
PROD	x				x					x	x							x									x	x	x	x		x			x
ARP				x	x					x				x	x	x					x	x									x				x
JPetriNet				x	x				x													x				x									x
Petri .NET Simulator				x	x				x				x	x							x											x			
QPME	x		x		x			x	x					x						x			x			x	x	x	x			x	x		

The table above is the comparison between the Petri net tools surveyed in section III based on five main criteria. The first group of criteria is the Petri net supported. In this category, the Petri net tools are compared in terms of what type of Petri net is supported. Majority of the tools support Place/Transition Petri net with some supporting high-level Petri net (i.e. AlPiNA, CoopnBuilder, PROD, and QPME). However, QPME stands out in this category as it supports Queuing Petri net (a combination of Queuing Network and Petri net). For Continuous Petri net, only Snoopy and Charlie support it.

The second criteria of comparison are the components in each tool. Most of the tools provide a graphic editor and a fast simulation on Petri net. Tools that provide graphic editor and fast simulation on Petri net can be good teaching materials. Users will be able to edit the Petri net and simulate different Petri net to aid them in understanding Petri net. PEP has the highest number of components which includes graphical editor, state spaces, condensed state spaces, token game animation, place invariants, transition invariants, net reduction, model checking, Petri net generator and interchange file format. Users will be able to experience more on PEP compared to the other tools.

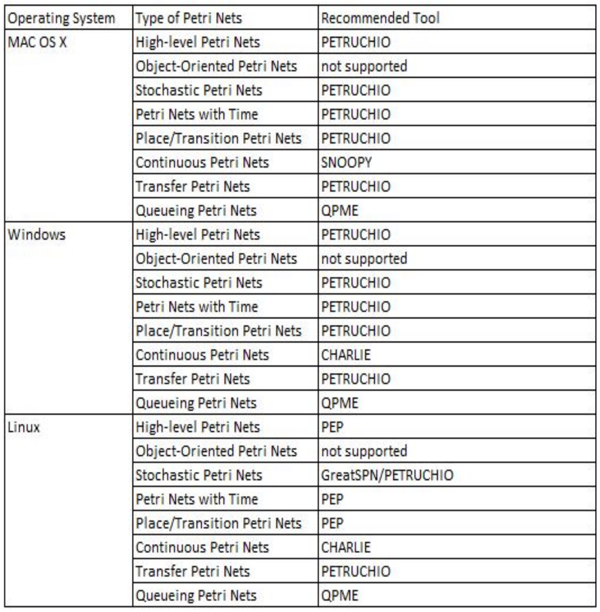
The next category of comparison is the analysis of Petri net. Some of the tools surveyed provide simple performance analysis, while tools such as GreatSPN, PEP, Charlie, JSARP, Xpetri, ARP and JPetriNet provide structural analysis. GreatSPN and QPME are also able to carry out advance performance analysis. The reachability graph based analysis is however only able to be carried out by MARCIE.

The next criterion used to compare between the Petri net tools is the environment. LoLA, PETRUCHIO and QPME have the highest amount of environment supported for their tools with six environments for each of them. Tools like AlPiNA, CoopnBuilder, MIST, Yasper, PAPETRI, ARP and Petri .NET Simulator is very environment specific with each of them only supporting one specific environment to be run on.

The final criterion of comparison is the pricing of the Petri net tools. All of the Petri net tools are either free of charge or free of charge for academic purpose to be downloaded.

In this section, Table-2 is presented to recommend Petri net tool for a few scenarios which includes the users' operating systems and the types of Petri net to be analyzed.

Table-2. Recommended Petri net tools for different usage.



Operating System	Type of Petri Nets	Recommended Tool
MAC OS X	High-level Petri Nets	PETRUCHIO
	Object-Oriented Petri Nets	not supported
	Stochastic Petri Nets	PETRUCHIO
	Petri Nets with Time	PETRUCHIO
	Place/Transition Petri Nets	PETRUCHIO
	Continuous Petri Nets	SNOOPY
	Transfer Petri Nets	PETRUCHIO
	Queueing Petri Nets	QPME
Windows	High-level Petri Nets	PETRUCHIO
	Object-Oriented Petri Nets	not supported
	Stochastic Petri Nets	PETRUCHIO
	Petri Nets with Time	PETRUCHIO
	Place/Transition Petri Nets	PETRUCHIO
	Continuous Petri Nets	CHARLIE
	Transfer Petri Nets	PETRUCHIO
	Queueing Petri Nets	QPME
Linux	High-level Petri Nets	PEP
	Object-Oriented Petri Nets	not supported
	Stochastic Petri Nets	GreatSPN/PETRUCHIO
	Petri Nets with Time	PEP
	Place/Transition Petri Nets	PEP
	Continuous Petri Nets	CHARLIE
	Transfer Petri Nets	PETRUCHIO
	Queueing Petri Nets	QPME

Table-2 summarizes on the recommended Petri net tools for different environments and different types of Petri net to be analyzed. Overall, users running on the MAC OS X are recommended to use the tool PETRUCHIO which supports most of the different types of Petri net. For Continuous Petri net, users will need to use the tool SNOOPY; while for Queuing Petri net, users are recommended to use the tool QPME. For Windows users, PETRUCHIO is recommended too as it supports most of the Petri net types. However, analysis of Continuous Petri net and Queueing Petri net requires users to use the tool CHARLIE and QPME respectively. For those running on Linux, there isn't any specific tool overall that can analyze most of the different types of Petri net. For High-level Petri net, Petri net with time and Place/Transition Petri net, users are recommended to use the tool PEP. For Stochastic Petri net, users are recommended to use GreatSPN or PETRUCHIO. CHARLIE is recommended for the analysis of Continuous Petri net while QPME is recommended for the analysis of Queueing Petri net.

Overall, the PEP tool offers the most components and analysis types amongst the twenty Petri net tools compared. However, due to the lack of supporting environment, PEP is not user friendly. Users will need to have Linux or Sun operating system to support the PEP tool.

Where else PETRUCHIO supports most of the major environments (MAC OS X, Windows, Linux), it is also the tool with the second highest component and analysis types. Based on Table-2, each operating system has at least two types of Petri net that is able to be analyzed by PETRUCHIO. From this deduction, we can conclude that PETRUCHIO is the best tool as it supports most of the operating system and it has an adequate amount of features in it.

REFERENCES

[1] Petri, C. A. 1962. Kommunikation mit automaten.

[2] Brauer, W., and Reisig, W. 2009. Carl Adam Petri and "Petri nets". Fundamental Concepts in Computer Science, 3: 129-139.

[3] Murata, T. 1989. "Petri nets: Properties, analysis and applications," Proceedings of the IEEE, 77(4): 541-580.

[4] Ameedeen, M. A. 2012. A model driven approach to analysis and synthesis of sequence diagrams (Doctoral dissertation, University of Birmingham).

[5] Trompedeller, M. 1995. A Classification of Petri Nets.

[6] Hostettler, S., Marechal, A., Linard, A., Risoldi, M., and Buchs, D. 2011. High-Level Petri Net Model Checking with AlPiNA. Fundamenta Informaticae, 113(3): 229-264.

[7] Al-Shabibi, A., Buchs, D., Buffo, M., Chachkov, S., Chen, A., and Hurzeler, D. 2003. "Prototyping object oriented specifications," In Applications and Theory of Petri Nets 2003 (pp. 473-482). Springer Berlin Heidelberg.

[8] Chiola, G., Franceschinis, G., Gaeta, R., and Ribaudo, M. 1995. GreatSPN 1.7: graphical editor and analyzer for timed and stochastic Petri nets. Performance Evaluation, 24(1): 47-68.

[9] Schmidt, K. 2000. "Lola a low level analyser," In Application and Theory of Petri Nets 2000 (pp. 465-474). Springer Berlin Heidelberg.

[10] Grahlmann, B., and Best, E. 1996. "PEP-more than a Petri net tool," In Tools and Algorithms for the Construction and Analysis of Systems (pp. 397-401). Springer Berlin Heidelberg.

[11] Heiner, M., Herajy, M., Liu, F., Rohr, C., and Schwarick, M. 2012. "Snoopy-a unifying Petri net tool," In Application and Theory of Petri Nets (pp. 398-407). Springer Berlin Heidelberg.

[12] Schwarick, M., Heiner, M., and Rohr, C. 2011, September. "Marcie-model checking and reachability analysis done efficiently," In: Quantitative Evaluation of Systems (QEST), IEEE 2011 Eighth International Conference on (pp. 91-100).

[13] Wegener, J., Schwarick, M., and Heiner, M. 2011. "A Plugin System for Charlie," In Proc. CS and P (pp. 531-554).

[14] Oliveira Lino, F.G. and Sztajnberg, A 2006 JSARP - Simulator and Analyzer Petri net in Java, a final project of the undergraduate course of Computer and Information Technology. Unpublished manuscript, University of Rio de Janeiro.

[15] Ganty, P., Van Begin, L., Delzanno, G., and Raskin, J-F. Coverability Checkers included in mist, online, URL: https://github.com/pierreganty/mist/wiki.

[16] Meyer, R., and Strazny, T. 2010, January. "Petruchio: From dynamic networks to nets," In: Computer Aided Verification (pp. 175-179). Springer Berlin Heidelberg.

[17] Riesz, M., Seckár, M., and Juhás, G. 2010. "PetriFlow: A Petri Net Based Framework for Modelling and Control of Workflow Processes," In: ACSD/Petri Nets Workshops (pp. 191-205).

[18] van Hee, K., Oanea, O., Post, R., Somers, L., and van der Werf, J. M. 2006, June. "Yasper: a tool for workflow modeling and analysis," In: Application of Concurrency to System Design, 2006. ACSD 2006. IEEE Sixth International Conference on (pp. 279-282).

[19] Berthelot, G., Johnen, C., and Petrucci, L. (1991, January). "PAPETRI: Environment for the Analysis of PETRI nets," In Computer-Aided Verification (pp. 13-22). Springer Berlin Heidelberg.

[20] Geist, R., Crane, D., Daniel, S., and Suggs, D. 1994, December. "Systems modeling with xpetri. In Proceedings of the 26th conference on winter simulation," Society for Computer Simulation International pp. 611-618.

[21] Varpaaniemi, K., Heljanko, K., and Lilius, J. (1997, January). "Prod 3.2 An advanced tool for efficient reachability analysis," In Computer Aided Verification Springer Berlin Heidelberg, pp. 472-475.

[22] Maziero, C. A. 1990. ARP: Petri net analyzer. Control and Microinformatic Lab., Federal Univ. Santa Catarina, Santa Catarina, Brazil.

[23] Azevedo. M. 2004. JPetriNet, Sapucai Valley University, Brazil, online, URL: http://jpetrinet.sourceforge.net/.

[24] Genter, G. Petri .NET Simulator, online, URL: http://www.petrinetsimulator.com/.

[25] Kounev, S., and Dutz, C. 2009. "QPME: a performance modeling tool based on queueing Petri Nets," ACM SIGMETRICS Performance Evaluation Review, 36(4): 46-51.