

A Study on Petri Net Supporting Tools for System Modeling and Analysis

Damián Varga

damian.varga@student.tuke.sk

*Faculty of Electrical Engineering and Informatics
Technical University of Košice, Košice, Slovakia*

Slavomír Šimoňák

slavomir.simonak@tuke.sk

*Faculty of Electrical Engineering and Informatics
Technical University of Košice, Košice, Slovakia*

Abstract

Within this work we analyze and compare ten Petri net supporting tools for system modeling and analysis. A short summary of several available overviews of Petri net tools is performed before the tools analysis itself to evaluate the current state in this field. Principal parts of the paper are focused on analysis and evaluation of the selected tools. We provide detailed and practical description of selection of up to date tools, their evaluation based on various criteria, and recommendations for several use cases.

Keywords: analysis, modeling, Petri nets, supporting tools, tool comparison

1. Introduction

Petri nets are a widely adopted formal method [1], [2] for modeling and analysis of various types of systems. Process of modeling using Petri nets is greatly supported by their simple visual representation. On the other hand, they provide powerful modeling and analytical options [3], where some of them are unique to Petri nets. Because of that, when working with Petri nets, a suitable tool should be picked. Therefore, it is useful to have an overview that would describe the Petri net tools to potential users. And even though there are some overviews available, they are often outdated [4], [5], or do not analyze every tool [6]. And because of that, new overviews are needed over time. Therefore, the objective of this paper is to create a new overview of such tools.

Because of their modeling and analytical power, there are several ways, in which Petri nets can be used. The tools in this work were chosen in a way that tries to include up-to-date, freely available tools with a broad variety of supported Petri net types and corresponding modeling and analysis options that can be used in scientific, educational or even professional environment. More details on this work can be found in the related thesis [7].

2. Petri Nets

Petri nets [8] were described by Carl Adam Petri in his thesis in 1962 for the first time. They are instances of bipartite directed graphs that contain two types of nodes – places and transitions that alternate each other. *Places* [8], [9] are used to represent states of the modeled system and are depicted as circles. *Transitions* are used to represent events that may change the current system state and are depicted as boxes or bars. In every Petri net, a set of places and a set of transitions are disjoint sets [10], which means that there is no object that can be a place and a transition at the same time. Places and transitions are connected to each other by *arcs* [11], represented by arrows. Every arc must connect exactly one place and one transition [8]. An arc may also have a positive integer labeled to it [12], expressing its *weight*. The last element that can be found in Petri nets, is a *token* [11] that moves around the net, representing changes of the system's state.

Tokens can be present only in places, never in transitions or arcs, and their movement must be only in the direction of the arcs. This movement is performed by firing transitions [12]. When a firing of a particular transition occurs, a token is removed from every input place and is put to every output place of that transition. If an arc has a weight labeled to it, the number of tokens removed from or added to a place is equal to that particular weight. That means that each input place of a particular transition must contain a number of tokens that is equal than or greater to the weight of the arc that connects that place to the transition before a firing of that transition occurs. If this condition is fulfilled, we say the transition is *enabled* [3].

Any sequential firing of transitions of a Petri net is called a *firing sequence* [12].

2.1. Petri Net Properties

There are two types of Petri net properties [8] – behavioral and structural. Behavioral properties depend on the initial marking of the net. They include *reachability*, *boundedness*, *liveness*, *coverability*, *reversibility*, *persistence*, *synchronic distance* and *fairness*. Structural properties are not dependent on the initial marking of the net, only on the topological structure of the net. They include *structural liveness*, *(complete) controllability*, *structural boundedness*, *conservativeness*, *repetitiveness*, *consistency* and *structural B-fairness*. Further information about these properties can be found in [8].

2.1.1. Property Analysis Tools

There are several ways to analyze some Petri net properties. For example, reachability and coverability can be analyzed using a *reachability tree* [12] that contains all reachable markings or a *reachability graph* [13] that can be created from a reachability tree by representing nodes with the same marking as a single node. Similarly, it is possible to create a *coverability tree* and a *coverability graph* [8].

Incidence matrix [8] $A = [a_{ij}]$ is an $n \times m$ matrix, where n represents the number of places and m the number of transitions. Each matrix entry is a difference of the

weight of the arc directed from a specific transition to a specific place and the weight of the arc directed from that place to that transition.

Invariants [12] can be divided into two groups – *S-invariants* (sometimes called *P-invariants*) and *T-invariants*. An S-invariant [8] is a vector y , consisting of m elements that is a solution of the equation $Ay = 0$. A T-invariant [8] is a vector x , consisting of n elements that is a solution of the equation $A^T x = 0$.

State equation [12] is an equation, which, given the current marking m_k , can predict the following marking m_{k+1} by the k -th firing, where $k \geq 0$.

Model checking [14] is a way of verifying system models. The procedure uses a logical formula φ and a structure M and tries to decide, if M satisfies φ . There are two types of model checking – LTL (or PLTL) [14], [15] and CTL [14]. Their respective grammars are defined e.g. here [16].

When analyzing Free Choice Petri Nets (defined in 2.2), there are two important types of place sets [9]. A *siphon* is a set of places that cannot get back tokens they lost. A *trap* is a set of places that cannot lose tokens, once they get them.

2.2. Petri Net Types

There are several types and extensions of Petri nets. The basic classification can be made by dividing Petri nets into three levels. Types of Petri nets that fall under these levels are listed on the Petri Nets World website [17].

In the remainder of this section, the most important Petri net types are mentioned and described.

Place/Transition (P/T) Petri Nets [17] are Petri nets, where places may contain several simple tokens and can represent integer values. They may also contain places with token capacities and arcs with weights.

State Machine [9] is a Petri net, where every transition has exactly one input place and one output place. *Marked Graph* is a Petri net, where every place has exactly one input transition and one output transition. *Free Choice Petri Net* is a Petri net, where every arc from a place is either the only output arc of that particular place, or the only input arc of a particular transition.

Colored Petri Net [18] is a Petri net, in which every token is assigned a specific color that can represent complex values. Each place and transition has a set of colors, it can interact with. A token color may change after a transition firing.

Workflow Petri Net [19] is a Petri net with only one starting and only one ending place.

There are several Petri net types that can operate with time. *Timed Petri Nets* [20] have a firing duration set for each transition. Some definitions [21] allow setting time durations to places and directed arcs, too. They can be divided into *Deterministic Timed Petri Nets* where those durations are either set, or there is at least an interval, which those durations must fall within, and *Stochastic Timed Petri Nets*, where those durations are randomized. *Generalized Stochastic Petri Nets* are Stochastic Timed Petri Nets that have an exponentially distributed random firing times set that coexist with firing delays. Those delays are deterministically equal to zero. As for Timed Petri Nets in this work, only timed transitions will be considered, if not stated otherwise.

Time Petri Nets [22] have a time interval set for each transition. The time this interval is associated with starts, when the transition that has this time assigned becomes enabled. A particular transition may be fired if enough time has elapsed, since it became enabled. This time, however, cannot go above the upper limit of the interval. *Timed-Arc Petri Nets* [23] are Petri nets, where every token has an age and arcs that go from a place to a transition have set time intervals that allow only tokens with an age in said interval to be moved.

In workflow Petri nets, a new property – *soundness* can be defined. This property is described in [24]. *Soundness* and *strong soundness* in relation to timed-arc Petri nets are described in [25].

2.3. Petri Net Markup Language

Petri Net Markup Language (PNML) [26] is an interchange format for Petri nets, based on XML. It is an international standard. Grammar of the language is available on the official website. The main point of PNML is to allow users to create compatible Petri net files with different tools [27]. For a tool to be considered a PNML supporting one [26], it has to be able to successfully import and export PNML files. Users may also create their own extensions of the language. Those extensions will, however, not be standard-compatible. Nowadays, there are many Petri net tools that accept PNML files [1].

2.4. Existing Overviews of Petri Net Tools

This section describes existing Petri net tools overviews. It will describe their methodologies, the amount of analyzed tools, their recency and their advantages and disadvantages.

2.4.1. Petri Nets Tool Database

This database [4] is very likely the largest overview that exists. It consists of approximately one hundred different tools. Information about the tools includes their homepage, availability, system environments, supported types of Petri nets, features, short tool description and contact details for authors of the tools.

The database also allows its users to view a table with an information summary of the tools and to add their own tool to it.

The biggest drawback of the database is the fact that, because the older tools are no longer reviewed, a lot of the tools are pretty old and, as a result, many times it is impossible to access their homepages.

2.4.2. A Survey of Petri Net Tools

This survey [11] is one of the shorter ones. Tools mentioned there are described very briefly. On the other hand, there is a decent amount of the tools mentioned in the paper. And, what is more, it is still possible to access most of the tools.

There are two tables at the end of the paper. One of them contains a summary of all tools and information about them – supported types of Petri nets, features, environments, they can run on and whether or not they are free of charge. In the other table, there is a list of recommended tools that are mentioned in the survey, for users' operating system and a required Petri net type.

The paper also chose the best tool of the analyzed ones – PETRUCHIO. The homepage of this tool is, however, no longer accessible.

2.4.3. Petri Net Tools – A Comparative Study

This study [5] is one of the longer ones. Before the descriptions of the analyzed tools are presented, the study describes its methodology and a tabular summary of all the tools.

The study evaluates technological and user costs and advantages of the tools. It evaluates the following: analysis options, net dynamisation, generation, user interface, available file formats for export, import, default file format, documentation and other sources of information about the tools, their experience and how the tools may be applied or combined.

The largest drawback of this study is that it is considerably old. It was written in 1996. It is, however, at least possible to take a look at its methodology of analysis and comparison.

2.4.4. A Comparison of Petri Net Simulation Tools

This paper [6] is one of the newer ones. It analyzes three Petri net tools – Pipe 2, CPN Tools and HP Sim. What is more, all three tools are still available at the moment. The overview provides information about the tools, about supporting types of Petri nets, their components and features, required system environments and evaluates their user interfaces.

The biggest disadvantage of this paper is that there are not many tools analyzed. This may not be sufficient for some readers. However, if one does not have very specific requirements while searching for a Petri net tool, this overview may be the one for them, because in others they might have too many options.

2.4.5. Demonstration Tools for Petri Nets

This thesis [28] analyzes six different tools. However, it is designed to find the best tool for educational use. The thesis provides the following information about the tools – origin information, supported Petri net types, distribution, system requirements, documentation, user interface and features of the tools.

The biggest advantages of this thesis are its clarity, because it is easy to find necessary information, thanks to clear paragraph segmentation, the fact that the descriptions contain practical information and screenshots of user interfaces.

A disadvantage of this thesis, as a Petri net tools overview, is that the tool summary and the information about the tools are not placed together. This can make navigating around the document and connecting information a bit more challenging.

2.4.6. Summary

One can say that there are quality Petri net tools overviews, although with some flaws. Every one of the overviews, however, had some qualities. If the quality is not information about the analyzed tools, it is at least an analysis and comparison methodology.

3. Tools Overview

The next part is an overview of various Petri net tools. It describes the tools and points out their advantages and disadvantages.

There are following information stated about the tools in this overview: supported Petri net types, available features, system requirements, user interfaces, experience, while using the tools, etc. There results of the review are summarized in tabular form at the end of the paper.

3.1. CPN Tools

CPN Tools [29] is a tool for colored and timed Petri net modeling, simulation and analysis.

When it comes to the Petri net analysis, the tool is able to analyze boundedness, liveness, fairness and perform home state analysis. To view the results, however, it is necessary to generate a file that contains them. The tool also contains state space analysis options [30] that contain an option to generate reachability graphs [29], [31].

When it comes to working with time in CPN Tools, it is possible to create timed colored Petri nets [29] that use timed color sets. Using these color sets, it is possible to set a time stamp to each token and restrict their movement based on a time counter. Time attributes for transitions and arcs can be simple time delays, or in case of arcs that go from a place to a transition, an amount of time that transition can fire earlier, but these attributes may be more complicated expressions, as well. As an example, it is worth mentioning random time delays from a specified interval, or an option to pick a time delay from a set interval [32].

CPN Tools also contains a variety of performance analysis tools and options [29] that include the ability to collect statistical data during simulations, such as an amount of tokens in a place or how many times a particular transition occurs in a simulation. These data may be saved to a file afterwards.

Other functionalities [29] of CPN Tools include the ability to perform model checking using a CTL-like temporal logic, write queries using an ML language and view net statistics [32].

The biggest disadvantage of CPN Tools is that the tool is very complicated to use. Even learning the bare minimum may take the whole day [6] and mastering the tool

much longer. Therefore, it is recommended just to be patient, research information online and experiment with the tool.

It should be noted that development on CPN Tools has been stopped recently [29] and a new tool – CPN IDE [34] has replaced this tool.

CPN Tools is available under GNU GPL v2 license [29].

3.2. Time petri Net Analyzer

Time petri Net Analyzer (TINA) [35] is a tool that, as the name suggests, primarily used for modeling, simulation and analysis of time Petri nets.

When it comes to Petri net analysis, one can analyze boundedness, liveness, reversibility, consistency, create reachability, coverability and marking graphs, perform structural analysis and reachability checking using TINA. The tool also allows performing model checking, as well as other options using commands. A list of command and model checking manuals is present on the official webpage [35].

The analysis results can be exported to various file formats. All of them, except ".ktz", can be basically interpreted as ordinary text files.

Disadvantages of TINA include not very smooth arc design and inability to display most of the special characters. A temporary solution for the latter is copying them from elsewhere and pasting them into the editor. However, after closing and re-opening the application, those characters no longer appear on screen.

In the file README.txt is some information about licensing, or availability, but not the specific license. After communication with Dr. Bernard Berthomieu, it was discovered that TINA is not available under a specific license, but it is distributed as a free-ware.

3.3. TAPAAL

TAPAAL [36], [37] is a tool for modeling, simulation and analysis of timed-arc P/T Petri nets developed at the Aalborg University in Denmark.

It is necessary to have Java runtime Environment v. 11.0 or higher [36] installed to be able to use the tool. However, on some higher versions it is possible that the tool will not run, so it is recommended to have exactly version 11.0 installed.

The tool offers various ways of analyzing Petri nets that include deadlock analysis, analyze k -boundedness for a specific k entered by user, compute minimal possible k , and analyze workflow nets – identify their starting and ending places, determine their soundness, strong soundness. The tool also allows its users to perform model checking, view net statistics and automatically change the net design based on user entered criteria.

An interesting feature of TAPAAL is an option to use Automatic Net Layout, which is an automatic way of changing a shape of a net using criteria set by a user.

The tool parts are licensed differently [36]. The user interface is available under Open Source License 3.0, but other parts are under BSD, or GNU GPL v2 licenses.

3.4. Petri Net Simulator

This tool [38] is a simple web application. It can only model and simulate P/T Petri nets and it does not offer any analysis options or PNML support.

Execution of modeling and simulation is very simple, however, if necessary, there is a video [39] that explains usage of this tool.

One of the biggest disadvantages of this tool is that there is no way to choose, which transition will be fired next. The tool fires all available transitions instead. This may cause unexpected behavior during the simulation of some models. Another drawback is that there is no way to move multiple net components at once. However, if there is a situation, in which it is necessary to extend the drawing board, it is possible to do so by making the webpage smaller by zooming out.

Because the tool is a web application, it does not have to be downloaded and installed. Therefore, it is relatively easy to access it.

3.5. Platform Independent Petri net Editor 2

Platform Independent Petri net Editor 2 (PIPE2) [40], [41], [42] is a tool used for modeling, simulation and analysis of P/T, generalized stochastic and colored Petri nets [40], [43]. It was created as a group project on Imperial College London [40].

The nets, created with this tool, are automatically saved as PNML files in a XML format. However, it is possible to export the nets to PNG format [40].

The tool offers several Petri net analysis options. It can determine whether or not a created net is a marked graph, a state machine, or some other type. The tool offers deadlock and invariant analysis, can calculate minimal siphons and traps and create a reachability/coverability graph and an incidence matrix.

PIPE2 is able to analyze liveness and boundedness of the network, as well, using positive T-invariants. When it comes to the liveness analysis, if the invariants are not used for the analysis, only existence of deadlocks can be checked.

The tool also offers performance analysis and computation of time and state metrics and analysis functions for generalized stochastic Petri nets.

When it comes to colored Petri nets, it is necessary to click the "Specify tokens" icon in the menu, represented by the palette, or press keys Ctrl, Shift and T simultaneously in order to set token colors and manually set arc attributes in order to enable transitions [43].

The latest version is version 5 [43], however, the recommended version is version 4 [40], [43]. The newest version [43] does not offer most of the modules the older one does. Version 5 is available under MIT License [43], version 4 is available under Open Software License 3.0 [44].

3.6. WOLFGANG

This application [45] supports modeling, simulation and analysis of P/T and colored Petri nets. Even though the developers of the tool mention only Java Runtime Environment 1.8 as a necessary system prerequisite [46], it is possible to run the tool

using a higher version, as well, as proven, when the tool successfully ran using OpenJDK 11 and 16.

When it comes to analysis options, the tool is able to analyze boundedness, although it is not possible to compute the number k , determine if a created net is an instance of a workflow net and if it is, it can determine the soundness of it.

Arcs are created by pressing a center of a place or a transition [45]. When creating a new place or a transition, it is possible to either create a new instance from the menu, or to place an arc into an empty space. This feature may rapidly increase modeling speed after a bit of training. It should be also said that when modeling a net, it is recommended to decrease the default size of places and transitions.

The biggest advantage of the tool is the simple way of modeling and analysis of colored Petri nets – setting colors, weights of arcs, or the fact that modeled nets have a very clear design. The tool is available under BSD 3-Clause license [46].

3.7. Snoopy

Snoopy [47], [48] is a tool that supports P/T, colored, time, timed, stochastic and other various kinds of Petri nets. The tool provides users with various modeling and simulation options that are offered based on a type of a modeled Petri net. It was developed at Brandenburg University of Technology, Cottbus-Senftenberg, Germany [49].

The tool is primarily used just for modeling and simulation of Petri nets [47]. For further analysis, tools Charlie and Marcie are recommended. However, as a part of the stochastic simulation, probabilistic LTL model checking can be performed, as described in [50]. Snoopy allows users to draw e. g. reachability graphs [48], too, but is unable to generate them itself.

For the work with colored Petri nets, it is better to read a manual [51], because the way of using colors in Snoopy is quite complex.

If needed, net attributes can be changed after a double click on them. Because of that, when it comes to arcs, it is more practical to draw them longer, or to drag any text further away from it, since the tool may detect the click as a click on that text.

Snoopy also offers a great variety of import and export options for the files.

When installing the tool, it is recommended to create an empty folder beforehand, because the tool will not do so during the installation.

3.8. Charlie

Charlie [49] is a tool that is used to analyze P/T Petri nets. It was created by developers of Brandenburg University of Technology, Cottbus-Senftenberg, Germany and Université Blaise Pascal, Clermont-Ferrand II, France.

The tool allows users to analyze files, created by Snoopy that were exported as ANDL or APNN files, and PNT files of the tool INA [49].

Charlie allows users to analyze structural boundedness, compute P and T invariants, incidence matrix, generate coverability and reachability graphs in graphic form, perform deadlock analysis or find siphons and traps. The tool also allows

analyzing nets using model checking. An interesting analysis function is the ability to find the shortest and the longest path in an analyzed net.

The application also features a table, containing analysis information about the net. The information in this table contain liveness, reversibility, k -boundedness, structural boundedness and conservativeness analysis, whether every transition (or place) has a pre or a post place (or a transition) and some more. Further information, including persistence check, can be found in the generated text protocol, as well.

It is recommended to create an empty folder first, when installing the tool, for the same reason as with Snoopy.

3.9. HPetriSim

This tool [52] is used to model and simulate P/T and timed Petri nets.

It is a modeling and simulation tool only, meaning that there is no way to perform complex analysis of created nets in this tool.

In the menu of the application, the most important options are in the menu "Format", while clicking a place, transition, or an arc. This is, for example, an option to make timed transitions, or to put tokens in places.

However, the biggest problem is that there is no way of choosing, what transition should be fired next. The program chooses it automatically. Because of that, when there is a need to choose a concrete transition from a group of enabled transitions to be fired, the tool may not work properly. Another drawback of this tool is that after a simulation, tokens remain graphically in the places, where they ended, but they are still present in their initial places in the menu "Format". The tool is available under GNU GPL v3.0 license [52].

3.10. WoPeD

WoPeD (Workflow Petri Net Designer) [53] is a tool created by the developers of Cooperative State University Karlsruhe, Germany. It is primarily used to model, simulate and analyze workflow Petri nets.

To run the tool, it is necessary to have Java Runtime Environment 11 or more installed [53].

Like in WOLFGANG, it is possible in WoPeD to create arcs, connected with places or transitions, by simply drawing an arc without connecting it to anything. Since arcs are drawn the same way, as in WOLFGANG, one could assume that while learning, how to use the tool, some unintended places and transitions would be drawn. However, this would most likely not be the case here, since the area, you need to click in order to create an arc, is very small.

For regular P/T Petri nets there are some analysis options in WoPeD. The tool is able to generate net metrics or generate a coverability graph that may be exported as a PNG image. Reachable markings of the net can be displayed by clicking on the nodes of this graph. Other features include analysis of P-invariants, soundness, liveness and boundedness.

When it comes to workflow Petri nets, there are some interesting features present in this tool. It is possible to analyze a plan of real-life events, especially in business. Logic gates can be put in place of transitions. When there is a transition connected to two output places a probability of a system transferring to states, represented by those places, can be set. Every transition may also be given a message, a resource, or a time trigger. It is also possible to compute expected length and resources of every action of a set plan and to set a role to each of those actions. Sound workflow Petri nets may also be converted into a textual description and vice versa. The tool is available under GNU Lesser General Public License [53].

4. Evaluation

This section serves as a brief summary and evaluation of all analyzed tools. It mentions some information about the tools, mostly using tables. It contains recommendations about the usage of the applications, as well.

4.1. Supported Petri Net Types

The Table 1 shows, what types of Petri nets are supported by the tools. In order for a tool to be counted for the column labeled as "Time", a tool needs to support at least one of Petri net types that use time, as described in section 2.2.

	P/T	Time	Colored
CPN Tools	✓	✓	✓
TINA	✓	✓	
TAPAAL	✓	✓	
Petri Net Simulator	✓		
PIPE2	✓	✓	✓
WOLFGANG	✓		✓
Snoopy	✓	✓	✓
Charlie	✓		
HPetriSim	✓	✓	
WoPeD	✓		

Table 1. Supported Petri net types.

The only three tools that support all Petri net types in all three categories are CPN Tools, PIPE2 and Snoopy. Snoopy, however, can support many more types, making it the most versatile tool in this regard.

4.2. Supported Operating Systems

The Table 2 shows, what operating systems are supported by every tool. Please note that the table does not specify concrete operating system versions, so the tools may not work properly on every device, especially, when using an older version of an operating system.

When it comes to Petri Net Simulator, there is no official information on supported operating systems. Since the tool is available online, the whole row is filled with the word "internet".

Most of the tools support all three biggest computer operating systems. Only CPN Tools and HPetriSim support only Windows. It should be noted, however, that the CPN Tools does not support Mac OS X as a whole, but the simulator in the tool is supported for this operating system [29]. This fact is represented by the "*" character in the table. The simulator runs on Linux, too, but this version is unsupported [29], so it is not considered for this table.

	Windows	Linux	Mac OS	Solaris
CPN Tools	✓		*	
TINA	✓	✓	✓	✓
TAPAAL	✓	✓	✓	
Petri Net Simulator	internet	internet	internet	internet
PIPE2	✓	✓	✓	
WOLFGANG	✓	✓	✓	
Snoopy	✓	✓	✓	
Charlie	✓	✓	✓	
HPetriSim	✓			
WoPeD	✓	✓	✓	

Table 2. Supported operating systems [7].

4.3. Supported Petri Nets Properties and Functions

The table 3 shows analysis of what properties is possible in the analyzed tools.

	Rea	Bnd	Liv	Cov	Snd	Rev	Csv	Prs	Cst	Frn
CPN Tools	✓	✓	✓			✓*				✓
TINA	✓	✓	✓	✓		✓			✓	
TAPAAL		✓	✓*		✓s					
Petri Net										
PIPE2	✓	✓	✓	✓						
WOLFGANG		✓			✓					
Snoopy										
Charlie	✓	✓S	✓	✓		✓	✓	✓		
HPetriSim										
WoPeD	✓	✓	✓	✓	✓					

Table 3. Supported Petri net properties. The acronyms in the table have the following meaning: Rea – reachability, Bnd – boundedness, Liv – liveness, Cov – coverability, Snd – soundness, Rev – reversibility, Csv – conservativeness, Prs – persistence, Cst – consistency, Frn – fairness.

Please note that only automated analysis is counted for this table, meaning that e. g. manual modification of place capacity, to make a net k -bounded is not counted. Analysis options using model checking are excluded from the table, as well.

When it comes to analyzing Petri net properties, Charlie could analyze most of them. Its ability to analyze structural boundedness is noted by the letter "S" in the table, whereas TAPAAL's ability to analyze strong soundness is noted by the letter "s". The star characters in the table represent the fact that when it comes to reversibility, CPN Tools can perform home state analysis and when it comes to liveness, TAPAAL can analyze deadlock existence.

When it comes to model checking, the tools that allow users to perform it are TINA, TAPAAL, Charlie and Snoopy, but Snoopy offers model checking only for a fraction of supported Petri net types. However, files, created with this tool, may be exported to other tools that support model checking [54].

The other features worth pointing out are invariant analysis, supported by TINA, PIPE2 and Charlie, and siphon/trap analysis, supported by PIPE2 and Charlie.

There are, obviously, more features that these tools can offer. Those have been mentioned in detailed analysis for each tool. For this summary, only the most important features have been picked.

4.4. PNML Support

PNML is supported by the following of the mentioned tools:

- CPN Tools,
- TINA,
- TAPAAL,
- PIPE2,
- WOLFGANG,
- Snoopy,
- WoPeD.

When looking at the list, it could be assumed that at least every P/T Petri net, exported as a PNML file, created by any one of these tools, could be opened in the others without issues. But that is not always the case. When using CPN Tools, the tool could not open other tools' files and when trying to open PNML files created by CPN Tools, only WoPeD managed to do it, but the text of the net was grouped together in one area. Problems that arose while importing PNML files to Snoopy from other tools were mainly with the net shape and problems while importing to WoPeD were mostly with text that behaved like described in the example with CPN Tools. When trying to import Snoopy's and WoPeD's files elsewhere, there was not a single attempt that would be entirely successful, too. Snoopy's PNML files lacked layout information. TAPAAL handled this task the best, using its Automatic Net Layout feature. WoPeD's PNML files were successfully opened only by PIPE2, WOLFGANG and Snoopy, but the text in PIPE2 and WOLFGANG was scattered, the marking in PIPE2 and Snoopy was not present and Snoopy did not preserve the correct shape of the net.

When it comes to PIPE2, to import/export PNML files from/to other tools, it is necessary to change the file type from .pnml to .xml and vice versa. The only

successful import to this tool was using WoPeD's file which was described earlier. As for exports, files from PIPE2 were successfully opened by Snoopy, but the file lacked marking and was not in the correct shape, and WoPeD, but the tokens and the text disappeared.

It is also important to state that the import of PNML files created by WoPeD and PIPE2 to Snoopy works only using the "Import High-Level Petri Net PNML" option.

When it comes to exports of TINA's PNML files, apart from the problems mentioned earlier, wrong text appeared in TAPAAL and Snoopy. Export to WOLFGANG was a total success. As for imports, TAPAAL's and WOLFGANG's PNML files were imported successfully and Snoopy's file was shown only as text. Imports of other files were not successful.

Using TAPAAL and WOLFGANG, it is possible to perform a successful PNML import and export between these two tools. Problems that arose while performing import to and export from other tools, while using this duo of tools, were covered in the earlier part of this subsection.

4.5. Recommendations

Obviously, not every tool is suitable for every purpose. This part will, therefore, try to recommend tools for different purposes.

For scientific use, the best tools are CPN Tools, TINA, PIPE2, Snoopy and Charlie. CPN Tools and TINA can analyze a broad variety of analysis options, but the tools are not that easy to handle. PIPE2 is very straightforward to use and contains many analysis options. A combination of Snoopy and Charlie may seem complicated at first, but thanks to a massive pool of features it is most likely the best fit for scientific research. Other suitable tools include TAPAAL and WoPeD, thanks to a decent amount of analysis options.

For high school students, the best tools may be PIPE2 and TAPAAL, due to the fact they are not difficult to use and may be used to analyze Petri nets, and Petri Net Simulator because of its simplicity. Other suitable tools are TINA, WOLFGANG and WoPeD.

With all of this said, it is possible that sometimes it will be necessary to use multiple tools at once, because there is simply not a perfect, universal tool containing everything.

4.6. Comparison to Existing Overviews

The main goal of this paper was to create an overview that would contain currently available tools. Obviously, this goal has been successfully achieved.

One of the advantages of this paper is that there was some practical experience mentioned when describing most of the tools. These things are usually absent in most of other overviews.

In "Demonstration Tools for Petri Nets" [28] and "A Comparison of Petri Net Simulation Tools" [6] are some of those facts mentioned, but most of the tools that are mentioned here, are not present in those overviews. In "Petri Net Tools – A

Comparative Study" [5] is not mentioned even one of the tools, analyzed here. And the official database [4] and "A Survey of Petri Net Tools" [11] contain a very limited amount of information on user experience while working with the tools or tips for using them. Therefore, it is possible to say that this paper brings new information to the table.

If there is a need to look up information about a tool, not mentioned here, it is recommended to check out the official database [4]. It is very likely that the tool will be mentioned there.

5. Conclusion

To make the process of modeling and analysis using Petri nets easier, it is very useful to know, what tool is appropriate for this task. Whether a simple interface is enough, or complicated analysis options are necessary. That is what tools overviews are for. And, as it was stated, there are quite a few of them, but there is not a perfect one. Therefore, this overview is just a small part of the group of all overviews.

As the time goes, this overview will become more and more out of date. When the tools here will not be used anymore, because they will be replaced by newer tools, this review will not be needed anymore. Until then, however, this paper may serve as a source of relevant information.

In the future it would be possible to extend this work by additional analysis of selected tools or simply by analyzing other tools that were not included in this paper.

References

- [1] S. Šimoňák and M. Šolc, "Enhancing Formal Methods Integration with ACP2Petri," *Journal of Information and Organizational Sciences*, vol. 40, no. 2, Dec. 2016. [Online]. Available: <https://www.jios.foi.hr/index.php/jios/article/view/1063>. [Accessed: Aug. 20, 2022].
- [2] S. Šimoňák, M. Hudák, and Š. Korečko, "APC Semantics for Petri Nets," *Informatika*, vol. 32, no. 3, pp. 253–260, 2008. [Online]. Available: <https://www.informatika.si/index.php/informatika/article/viewFile/197/194>. [Accessed: Aug. 20, 2022].
- [3] S. Šimoňák and M. Tomášek, "ACP Semantics for Petri Nets," *Computing and Informatics*, vol. 37, no. 6, pp. 1464–1484, 2018. [Online]. Available: https://www.cai.sk/ojs/index.php/cai/article/view/2018_6_1464. [Accessed: Aug. 20, 2022].
- [4] University of Hamburg, Germany. Complete Overview of Petri Nets Tools Database. [Online]. Available: <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html>. [Accessed: Jun. 7, 2022].
- [5] D. Wikarski (ed.), "Petri net tools: a comparative study," Fraunhofer-Institut fuer Software- und Systemtechnik, Berlin (Germany), p. 68, 1996.

- [Online]. Available: <https://www.th-brandenburg.de/~wikarski/dokumente/pntstudy.ps>. [Accessed: Jun. 7, 2022].
- [6] J. P. L. Petter and S. Sawicki, "A COMPARISON OF PETRI NET SIMULATION TOOLS," In Proc. XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA, 2017, pp. 26 – 27. [Online]. Available: <https://publicacoeseventos.unijui.edu.br/index.php/salaconhecimento/artic le/view/8273/6999>. [Accessed: Jun. 7, 2022].
- [7] D. Varga, "Petri Net supporting tools for modeling and analysis of systems" (in Slovak), Bachelor thesis, Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, Slovakia, 2022, Supervisor: S. Šimoňák. [Online]. Available: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=B599B75453D2B93F16D E3075B660>. [Accessed: Jul. 12, 2022].
- [8] T. Murata, "Petri nets: Properties, Analysis and Applications," In Proc. *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989. [Online]. Available: <https://inst.eecs.berkeley.edu/~ee249/fa07/discussions/PetriNets-Murata.pdf>. [Accessed: Jun. 8, 2022].
- [9] S. Šimoňák, "Modeling and analysis of systems based on formal methods integration" (in Slovak), Habilitation thesis, Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, Slovakia, 2018. [Online]. Available: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=78AFF12394AFE35F8D1 F7C9F885D>. [Accessed: Jun. 7, 2022].
- [10] N. V. Nenkov and E. Z. Spasova, "Implementation of a Neural Network Using Simulator and Petri Nets," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 412–417, Jan. 2016. [Online]. Available: <https://dx.doi.org/10.14569/IJACSA.2016.070155>. [Accessed: Jun. 7, 2022].
- [11] W. J. Thong and M. Ameen, "A Survey of Petri Net Tools," *Lecture Notes in Electrical Engineering*, vol. 315, pp. 537–551, Jan. 2015. [Online]. Available: https://dx.doi.org/10.1007/978-1-61779-361-5_21. [Accessed: Jun. 7, 2022].
- [12] A. Venkateswarlu and A. Halder, "A Study of Petri Nets Modeling, Analysis and Simulation." Kharagpur – 721302, West Bengal, India: Department of Aerospace Engineering, Indian Institute of Technology Kharagpur, Aug. 2006. [Online]. Available: <https://abhishekhalder.bitbucket.io/PetriNetReport.pdf>. [Accessed: Jun. 7, 2022].

- [13] X. Ye, J. Zhou, and X. Song "On reachability graphs of Petri nets," *Computers and Engineering – CEE*, vol. 29, pp. 263–272, Mar. 2003. [Online]. Available: <https://repository.um.edu.mo/handle/10692/7875>. [Accessed: Jun. 7, 2022].
- [14] M. Müller-Olm, D. A. Schmidt, and B. Steffen, "Model Checking – A Tutorial Introduction," In Proc. *Static Analysis: 6th International Symposium, SAS'99, Venice, Italy, September 22-24, 1999, Proceedings*, vol. 1694. Springer, 1999, pp. 330–354.
- [15] J. Strejček, "Linear Temporal Logic: Expressiveness and Model Checking," Dissertation, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2004. [Online]. Available: <https://www.fi.muni.cz/~xstrejc/publications/thesis.pdf>. [Accessed: Jun. 7, 2022].
- [16] A. Gurfinkel, "MODEL CHECKING," Carnegie Mellon University, Computer Science Department, 2014. [Online]. Available: <https://www.cs.cmu.edu/~emc/15414-s14/lecture/ModelChecking.pdf>. [Accessed: Jun. 7, 2022].
- [17] University of Hamburg, Germany. A Classification of Petri Nets. [Online]. Available: <https://www2.informatik.uni-hamburg.de/TGI/PetriNets/classification/>. [Accessed: Jun. 7, 2022].
- [18] J. Wang and W. Tepfenhart, *Colored Petri Nets*, June 2019, pp. 271–289. [Online]. Available: <https://dx.doi.org/10.1201/9780429184185-10>. [Accessed: Jun. 7, 2022].
- [19] G. Juhás, "Detekcia uviaznutí v diskrétnych udalostných systémoch so zdieľanými zdrojmi," [Online]. Available: https://uim.fei.stuba.sk/wp-content/uploads/2018/02/masus_part_one.pdf. [Accessed: Jun. 7, 2022].
- [20] L. Popova-Zeugmann, "Timed Petri Nets," in *Time and Petri Nets*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 139–172. [Online]. Available: https://doi.org/10.1007/978-3-642-41115-1_4. [Accessed: Jun. 7, 2022].
- [21] J. Wang, *TIMED PETRI NETS: Theory and Application*. Miami, FL: Springer Science + Business Media, LLC, 1998.
- [22] B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. Roux, "Comparison of the Expressiveness of Timed Automata and Time Petri Nets," In Proc. FORMATS 2005 - 3rd International Conference on Formal Modeling and Analysis of Timed Systems, Sep. 2005, pp. 211–225. [Online]. Available: https://doi.org/10.1007/11603009_17. [Accessed: Jun. 7, 2022].
- [23] L. Jacobsen, M. Jacobsen, M. H. Møller, and J. Srba, "Verification of Timed-Arc Petri Nets," in *SOFSEM 2011: Theory and Practice of Computer Science*, I. Černá, T. Gyimóthy, J. Hromkovič, K. Jefferey, R.

- Královič, M. Vukolić, and S. Wolf, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 46–72. [Online]. Available: <https://homes.cs.aau.dk/~srba/files/JJMS:SOFSEM:11.pdf>. [Accessed: Jun. 7, 2022].
- [24] P. E. Hoffmann, "Workflow Nets: Reduction Rules and Games," Dissertation, Faculty of Mathematics, The Technical University of Munich, Munich, Germany, 2017. [Online]. Available: <https://dnb.info/1156713463/34>. [Accessed: Feb. 26, 2023].
- [25] P. G. Jensen, J. A. Mateo, M. G. Sørensen, J. Srba, "Techniques and Tools for the Analysis of Timed Workflows," Department of Computer Science, Aalborg University, Aalborg, Denmark. Oct. 2015. [Online]. Available: <http://icetcs.ru.is/nwpt2015/SLIDES/Jiri.pdf>. [Accessed: Feb. 26, 2023].
- [26] LIP6. Presentation. [Online]. Available: <https://www.pnml.org/>. [Accessed: Jun. 7, 2022].
- [27] T. Shailesh, A. Nayak, and D. Prasad, "An UML Based Performance Evaluation of Real-Time Systems Using Timed Petri Net," *Computers*, vol. 9, no. 4, 2020. [Online]. Available: <https://www.mdpi.com/2073-431X/9/4/94>. [Accessed: Jun. 8, 2022].
- [28] W. de Zwijger, "Demonstration Tools for Petri Nets," Bachelor thesis, Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands, Mar. 2010. [Online]. Available: <https://theses.liacs.nl/pdf/10-05-WouterdeZwijger.pdf>. [Accessed: Jun. 7, 2022].
- [29] M. Westergaard and H. M. W. Verbeek. CPN Tools. [Online]. Available: <https://cpntools.org>. [Accessed: Jun. 8, 2022].
- [30] A. Cheng, S. Christensen, and K. H. Mortensen, "Model Checking Coloured Petri Nets – Exploiting Strongly Connected Components," *DAIMI Report Series*, vol. 26, no. 519, Mar. 1997. [Online]. Available: <https://tidsskrift.dk/daimipb/article/view/7048>. [Accessed: Jun. 8, 2022].
- [31] M. Szyrka. CP-net reachability graph generation in CPN Tools. [Online]. Available: https://alvis.kis.agh.edu.pl/wiki/pn2mc:cov_cpntools. [Accessed: Feb. 25, 2023].
- [32] M. Westergaard. Category: CPN Tools 4 Tutorial. [Online]. Available: <https://westergaard.eu/category/tutorials/cpn-tools-tutorial/>. [Accessed: Feb. 27, 2023].
- [33] S. Christensen and K. H. Mortensen, *Design/CPN ASK-CTL Manual*, Computer Science Department, University of Aarhus. [Online] Available: <http://cpntools.org/wp-content/uploads/2018/01/askctlmanual.pdf>. [Accessed: Jul. 12, 2022].

- [34] H. M. W. Verbeek and D. Fahland. CPN IDE. [Online]. Available: <https://cpnide.org/>. [Accessed: Feb. 8, 2023].
- [35] B. Berthomieu, F. Vernadat, and S. dal Zilio. Tlme petri Net Analyzer. Laboratoire d'analyse et d'architecture des systèmes/Centre national de la Recherche Scientifique. [Online]. Available: <http://projects.laas.fr/tina>. [Accessed: Jun. 8, 2022].
- [36] Aalborg University, Denmark. TAPAAL: Tool for Verification of Timed-Arc Petri Nets. [Online]. Available: <https://tapaal.net/>. [Accessed: Jun. 8, 2022].
- [37] A. David, L. Jacobsen, M. Jacobsen, K. Jørgensen, M. Møller, and J. Srba, "TAPAAL 2.0: Integrated Development Environment for Timed-Arc Petri Nets," In Proc. *Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'12)*, ser. LNCS, vol. 7214. Springer-Verlag, 2012, pp. 492–497.
- [38] I. Kim. Petri Net Simulator [Online]. Available: <http://petri.hp102.ru/pnet.html>. [Accessed: Jun. 8, 2022].
- [39] I. Kim. Petri Net modeling and simulation program. [Online]. Available: <https://www.youtube.com/watch?v=sMpqECG7-xA>. [Accessed: Jun. 8, 2022].
- [40] J. Bloom (et al.). Platform Independent Petri net Editor 2. [Online]. Available: <http://pipe2.sourceforge.net/index.html>. [Accessed: Jun. 8, 2022].
- [41] N. J. Dingle, W. J. Knottenbelt, and T. Suto, "PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, p. 34–39, Mar. 2009. [Online]. Available: <https://doi.org/10.1145/1530873.1530881>. [Accessed: Jun. 8, 2022].
- [42] P. Bonet, C. M. Llado, R. Puigjaner, and W. J. Knottenbelt, "PIPE v2.5: a Petri Net Tool for Performance Modeling," In Proc. *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*, San Jose, Costa Rica, Oct. 2007.
- [43] S. Tattersall, S. Doubleday, and G. Jones. Using PIPE 5. [Online]. Available: http://sarahtattersall.github.io/PIPE/user_guide.html. [Accessed: Sept. 5, 2022].
- [44] B. Kearns (et. al.). Platform Independent Petri Net Editor. [Online]. Available: <https://sourceforge.net/projects/pipe2>. [Accessed: Jun. 8, 2022].
- [45] IIG Telematics, University of Freiburg, "Wolfgang – The Petri Net Editor Documentation." [Online]. Available: <http://doku.telematik.uni-freiburg.de/wp-content/uploads/2015/05/wolfgang-userguide-doc.pdf>. [Accessed: Jun. 8, 2022].

- [46] IIG Telematics, University of Freiburg. WOLFGANG. [Online]. Available: <https://github.com/iig-uni-freiburg/WOLFGANG>. [Accessed: Jun. 8, 2022].
- [47] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick, "Snoopy – A Unifying Petri Net Tool," in *Application and Theory of Petri Nets*, S. Haddad and L. Pomello, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 398–407.
- [48] L. Popova-Zeugmann, "Time Petri Nets: Theory, Tools and Applications Part III," presented at 29th International Conference on Application and Theory of Petri Nets, Xi'an, China, 2008.
- [49] M. Heiner, M. Schwarick, and J.-T. Wegener, "Charlie – An Extensible Petri Net Analysis Tool," in *Application and Theory of Petri Nets and Concurrency*, R. Devillers and A. Valmari, Eds. Cham: Springer International Publishing, 2015, pp. 200–211.
- [50] M. A. Blätke, M. Heiner, and W. Marvan, "Tutorial - Petri Nets in Systems Biology," Otto-von-Guericke University Magdeburg, Magdeburg, Germany, Aug. 2011. [Online] Available: <https://dx.doi.org/10.13140/2.1.3796.6402>. [Accessed: Feb. 25, 2023].
- [51] M. Heiner, M. Herajy, F. Liu, and C. Rohr, *(Coloured) Hybrid Petri Nets in Snoopy - User Manual*, Brandenburg University of Technology Cottbus - Senftenberg, Mar. 2017. [Online]. Available: https://opus4.kobv.de/opus4-btu/files/4157/csr_01-17.pdf. [Accessed: Jun. 8, 2022].
- [52] H. Anschütz (et al.). HPetriSim. [Online]. Available: <https://github.com/Uzuul23/HPetriSim>. [Accessed: Jun. 8, 2022].
- [53] T. Freytag (et al.). Welcome to WoPeD. [Online]. Available: <https://woped.dhbw-karlsruhe.de/>. [Accessed: Jun. 9, 2022].
- [54] W. Marwan, C. Rohr, and M. Heiner, "Petri Nets in Snoopy: A Unifying Framework for the Graphical Display, Computational Modelling, and Simulation of Bacterial Regulatory Networks," *Methods in molecular biology (Clifton, N.J.)*, vol. 804, pp. 409–437, Jan. 2012. [Online]. Available: https://dx.doi.org/10.1007/978-1-61779-361-5_21. [Accessed: Jun. 9, 2022].