PNTools: a Multi-Language Environment to Integrate Petri Nets Tools

Allaoua Chaoui[1] and Ismail Hadjadj[2]

[1] Department of Computer Science, University Mentouri Constantine
Algeria, a_chaoui2001@yahoo.com

[2] Department of Computer Science, Mohamed KHIDER University- Biskra
Algeria, log.ismail@yahoo.fr

Abstract. The power of Petri nets resides on the availability of several tools for analysis and verification of properties of systems. Users have to combine the functionality of tools in different applications to take their advantages and overcome their disadvantages in order to solve complex problems that can't be solved using a single tool. This combination is difficult, because each tool has its own Petri net representation format. Most of those tools use an English interface, which penalizes its use by not Anglo-Saxon users. To eliminate these drawbacks, we propose a new multi-languages environment based on a standard file format (PNML file format), that integrates INA and PEP tools.

Keywords: Petri Nets, PNML, INA, PEP, Integration, Tools.

1 Introduction

Petri nets are a powerful graphical and mathematical tool for modeling, analyzing and verifying systems.

The use of computer-aided tools is a necessity for practical applications of Petri nets. Most Petri nets research groups have their own software packages and tools to assist the drawing, analysis, and/or simulation of various applications [1]. It is known that most of those tools use English interface. So, many obstacles face the users uninitiated in English. The solution of translating the tools becomes very expensive. To reduce the cost, we create a multi-languages environment based on French and English and simply extensible to many other languages; such as, Spanish, German and Italian…etc.

The variety of tools implies the persistence of various files formats. Users that want to combine different tools can't use the same file to analyze the net in different tools, because each tool had its own file format. The best way to solve this problem is the use of an interchange format that can be read by each of the used tools [2]. In the same context, an XML-based file format has been proposed in the International Conference on Application and Theory of Petri nets in 2000; it is the PNML [2].

1.1 Contribution

In this paper, we integrate two Petri nets tools: INA and PEP in one PNML-Based multi-languages environment called **PNTools** for (**P**etri Net **Tools**), developed using Java platform (Eclipse).

- The development of such tool has enabled us to overcome the perceived complexity of the problem of integration.
- We have integrated the tools: PEP and INA, in a very short time. And the results are very satisfactory.
- The possibility to choose the language is an additional advantage, allowing more flexibility when using our proposed software.
- The main contribution; is the gradual scalability. In fact, the integration of others Petri nets tools is possible, more simply and less costly. On the other hand, it is possible to add new languages (German, Italian, Spanish… etc.), this can be done, by injecting the interface translation into **PNTools**.

1.2 Paper organization

The remainder of this paper is organized as follows. In the second section, we review some previous related works. In section 3, we present the idea, design and implementation of our proposed **PNTools**. In section 4, we demonstrate our tool through a case study. Finally, section 5 concludes the paper and gives some perspectives of this research work.

2 Related Works

2.1 PNK

PNK or Petri net kernel [4] [5] [6] uses an English interface, developed originally under Python language (rebuild later under Java). PNK is not a Petri net tool in itself, but it's an environment which was developed to build applications for analyzing, modifying, simulating, or verifying Petri nets. The application functions which make up a PNK application use the interface of the Petri Net Kernel. In general, a PNK user only needs to implement some application functions and to put them together to an application [4].

The PNK can be used with any Petri nets type [7]. Because it is implemented to support PNML file format, both INA and LoLA [21] tools was integrated in PNK.
Users can create their own tools and integrate them in the PNK.

2.2 EZPetri

EZPetri [3][8] is an PNML-based environment developed under Eclipse platform and uses an English interfaces.

Figure 1 presents EZPetri architecture.

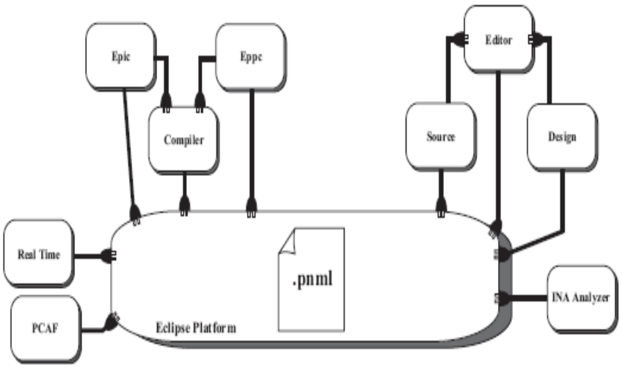


Fig.1. EZPetri architecture.

We saw that EZPetri consists of five plug-ins: the editor to edit nets, the compiler to interchange files formats, INA Analyzer to integrate INA tool, Real time to allow using timed Petri nets analysis and PCFA for power consumption analysis framework.

The integration of PEP in EZPetri confine in the import/export of net file format, but the PEP analyzing procedures are not integrated yet.

3 PNTools

We propose in the following our contribution to reduce the gap between members of Petri nets community which uses different Petri net types, tools and file formats [8]. **PNTools** uses a multi-languages interface. This increases the propagation of **PNTools** around the world, since it makes the use of the environment easier for a large range of users. Like the other tools, **PNTools** comes with English language as a default language; and French, Portuguese, Spanish are plugged-in.

As it is an integrate Petri nets tools environment, **PNTools** can also be used to create new tools because it is developed on an oriented object programming language Java [12] [15] which responds to the needs of reusability and portability.

To guarantee the extensibility and the flexibility of the **PNTools**, we implemented it on Eclipse [22] [23] platform. So, we can integrate other tools and languages as needed.

PNTools is not restricted to a single Petri net type. It can use a variety of Petri nets types, since it is based on PNML file format. The following figure shows the proposed **PNTools** architecture.

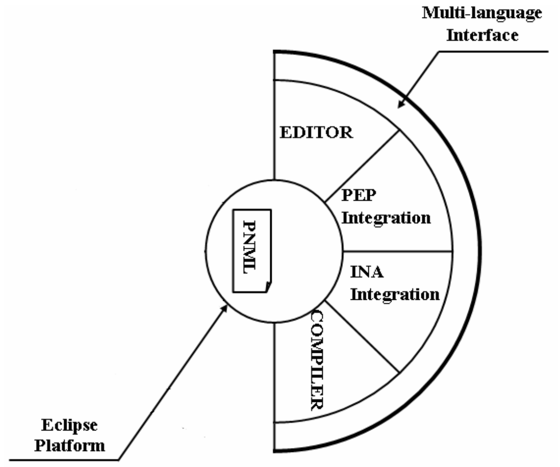


Fig.2. PNTools architecture

3.1 PNTools interface

The strategy of edition in the **PNTools** environment is a simple task.

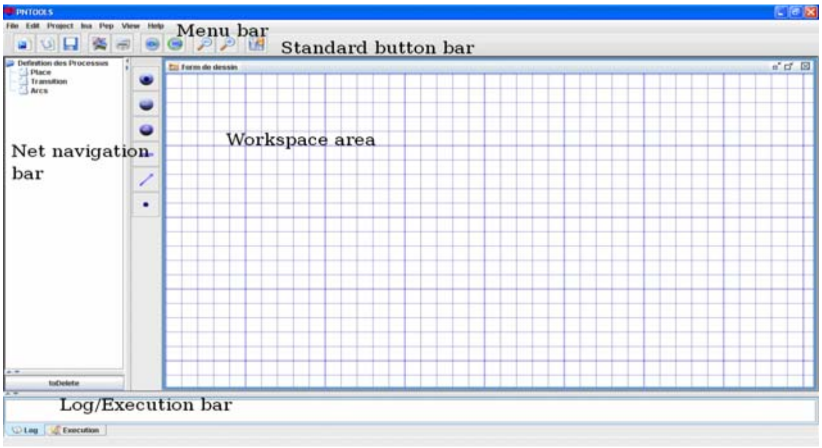


Fig.3. PNTools interface.

In **PNTools**, we reduce the number of areas and bars, to simplify the use of the interface. The design of **PNTools** divide the interface into five modules: a menu bar, a standard buttons panel, a workspace area, a net navigation panel and a log/execution bar (See figure 3).

The menu bar contains some essential functions such as create new project, save it … etc. The standard buttons panel holds shortcuts to the most used menu bar functions.

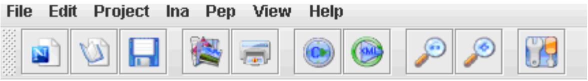


Fig.4. Menu bar and standard buttons panel.

Like a game, drawing nets; becomes very easy; it isn't an obligation to name places and transitions, a name is automatically attributed when an element is created. The grid in the workspace area helps users to arrange and align the net places and transitions. We can export the drawn net into a **.JPG** image file format.

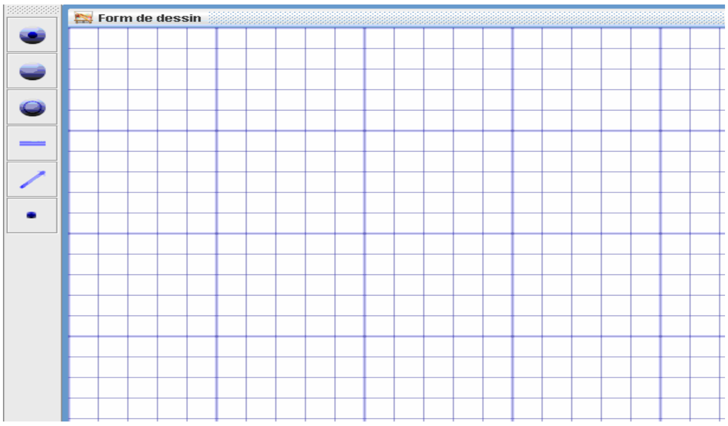


Fig.5. A part from workspace area.

The bar in the left of the windows (figure 6) is the net navigation panel; which contains the elements tree. It allows directly, selecting components and applying the pop up menu functions.

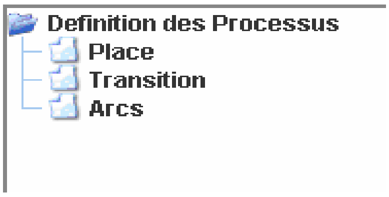


Fig.6. The net navigation bar.

Finally, the log/execution bar (Like it is shown bellow) is loaded to show the project session history.

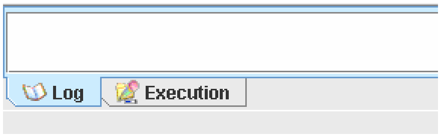
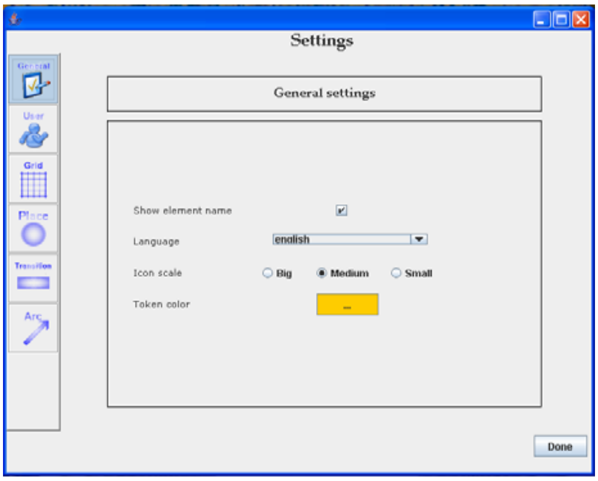


Fig.7. Log/execution bar.

3.2 Languages settings

To use the multi-language interface, a window of configuration is displayed when clicking the settings button in the standard buttons panel or from file/settings menu bar element.

The displayed window is shown in the following figure.



Settings

General settings

Show element name ☑

Language english

Icon scale ○ Big ● Medium ○ Small

Token color

Fig.8. Settings window.

Once displayed, we can choose the preferred language from a list. In fact, injecting a new language is not hard to do. It is only required to bring the translation file of the desired language into the installation directory.

In this window, we can specify the preferred net component options. We can also scale icons outlook for best view switching between three choices: Big, Medium and Small.

3.3 Compiler

The Petri Net Markup Language (PNML) is a Petri nets XML-based interchange format. In order to support different versions of Petri nets [9], **PNTools** uses this file format to swap inner files formats of INA and PEP.

Switching between file formats is available by:

- Exploiting one of the two formats ***.ll_net*** (See [10]) and ***.pnt*** (See [11]) in order to generate the final PNML file, or;
- Compiling the designed net (two tasks): firstly, **PNTools** verifies its integrity; after that, exports it as PNML file format, and;
- Translating the generated PNML file to each file format ***.ll_net*** or ***.pnt***.

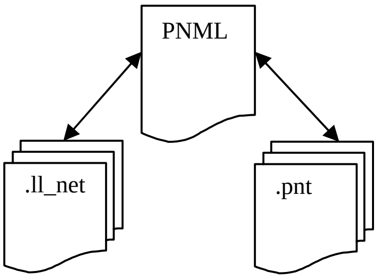


Fig.9. Interchange between file formats.

To produce the PNML file, we use the ***JDOM*** [12] library which is a standard interface for accessing and manipulating XML objects and their inter-relations exploiting XML functionalities.

3.4 Integrations

3.4.1 INA Integration

INA (Integrated Net Analyser) [11] tool was developed by Prof. Dr. Peter H. Starke in 1992. It is a tool shell that allow the interaction with user using a command line. It uses the English as interface inputs/outputs.

INA is an interactive menu-driven program that allows a user to edit (in a textual form), reduce, execute and analyze Petri nets models [14] (see figure 10).

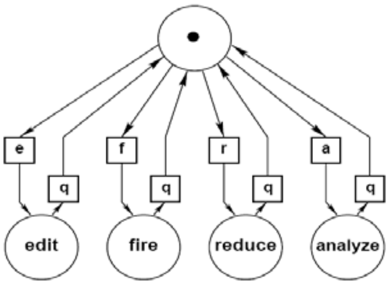
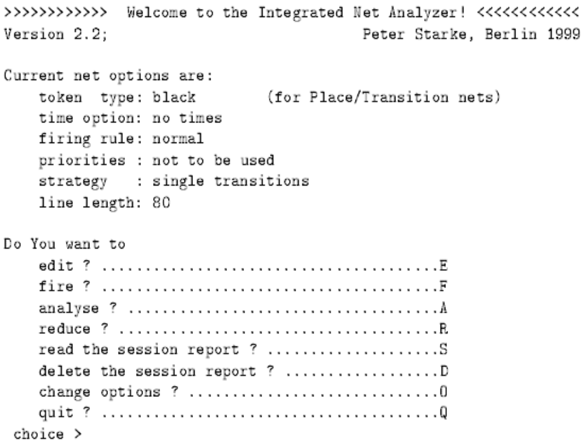


Fig.10. INA functions.

When we execute INA, the ***OPTIONS.ina*** file will be considered if it exists in the program directory. The ***COMMAND.ina*** file, in other hand, will be executed only if the answer of the question: " *Same procedure as last time?*" is <*Y*>. In this case, the stored commands of the last session are re-executed until <*H*> (for halt) is hold, or if all commands have been executed.

If there is no ***COMMAND.ina*** file, or the question above has been answered with <*N*>, the main menu appears on the screen and the selected options are displayed:



```
>>>>>>>>>>>>  Welcome to the Integrated Net Analyzer! <<<<<<<<<<<<
Version 2.2;                            Peter Starke, Berlin 1999

Current net options are:
    token  type: black        (for Place/Transition nets)
    time option: no times
    firing rule: normal
    priorities : not to be used
    strategy   : single transitions
    line length: 80

Do You want to
    edit ? ......................................E
    fire ? ......................................F
    analyse ? ...................................A
    reduce ? ....................................R
    read the session report ? ...................S
    delete the session report ? .................D
    change options ? ............................O
    quit ? ......................................Q
 choice >
```

Fig.11. INA execution.

The interaction between INA and **PNTools** is done by passing files as parameter. So, we took a long time to realized it because of the enclosed and encapsulated INA behavior, which hides the analysis process to users. In fact, INA is run in the background and detecting task of the generated files location is not so obvious.

To analyze the net, **PNTools** proceeds as follows:

- Create a file ***COMMAND.ina*** containing all chosen commands to run on the net,
- Lunch INA for the wanted analysis,
- Show the analysis results after exploiting the ***SESSION.ina*** file generated by INA.

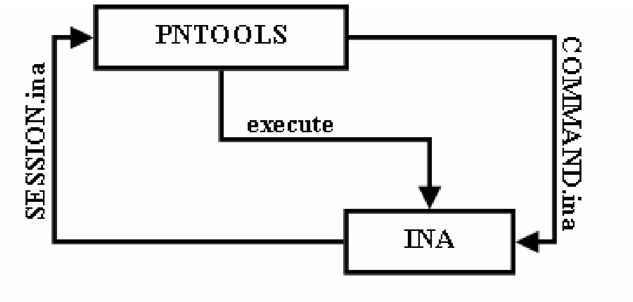
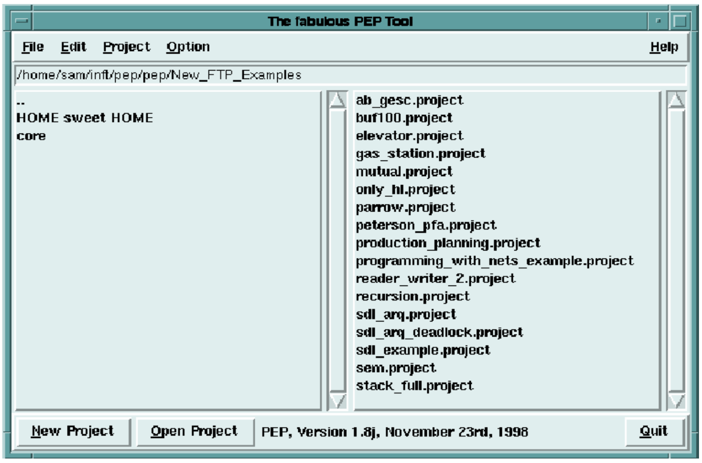


Fig.12. INA Integration.

3.4.2 PEP Integration

The PEP tool (Programming Environment based on Petri Nets) [16] [18] can be considered to be one of the most widely distributed Petri net based tools [17]. It supports the most important tasks of a good net tool, including HL and LL net editing and comfortable simulation facilities [18].

PEP has been implemented on Solaris 2.4, SunOS 4.1.3 and Linux. Executing PEP under Windows consumes a lot of time and effort; non specialist users can't execute this important tool on Windows. Because **PNTools** is developed under windows and our environment takes benefit from the portability of Eclipse platform, it can be executed on windows. For these reasons, we have chosen to integrate PEP in **PNTools**.



/home/sam/infl/pep/pep/New_FTP_Examples

..
HOME sweet HOME
core

ab_gesc.project
buf100.project
elevator.project
gas_station.project
mutual.project
only_hl.project
parrow.project
peterson_pfa.project
production_planning.project
programming_with_nets_example.project
reader_writer_2.project
recursion.project
sdl_arq.project
sdl_arq_deadlock.project
sdl_example.project
sem.project
stack_full.project

PEP, Version 1.8j, November 23rd, 1998

Fig.13. The start-up window of the PEP.

The PEP system ***bin*** directory contains subprograms; each one has a well determined task. Those subprograms can be executed as command lines under Windows. Therefore, the ***shell library*** [19] was imposed.

Forced by the PEP nature, we have proceeded differently; the interaction is done in two directions and **PNTools** analyzes the net as follows:

- From **PNTools** to PEP, we used an explicit parameter passing by variables. PEP Subprograms are launched sequentially and use the name of the file containing the net representation to be analyzed as attribute. Each subprogram creates a result file named ***SDTout***.
- In the opposite direction, the procedure is done through an exchange of files. Here **PNTools** gets results from ***SDTout*** file and displays the analysis results.

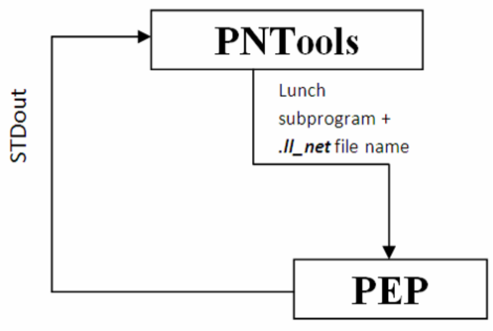


Fig.14: PEP Integration.

4 Case study

4.1 Petri net description [20]

In this example, the Petri net consists of two places (s1 and s2) and three transitions (t1, t2, and t3). The place s1 has one token. In modeling any general system, the places and transitions would be given some meaningful interpretation. For example, we have given the following interpretations to this net:

- s1: ready to accept coin
- s2: coin inserted
- t1: insert coin
- t2: accept coin
- t3: reject coin

According to this interpretation, the net below captures the behavior of a simple vending machine.

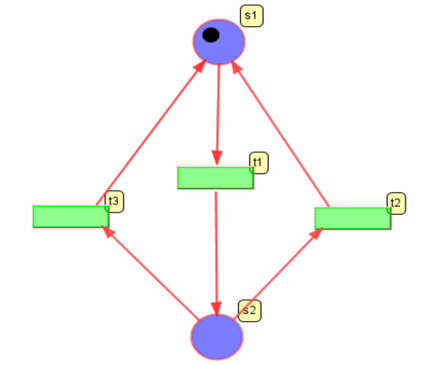
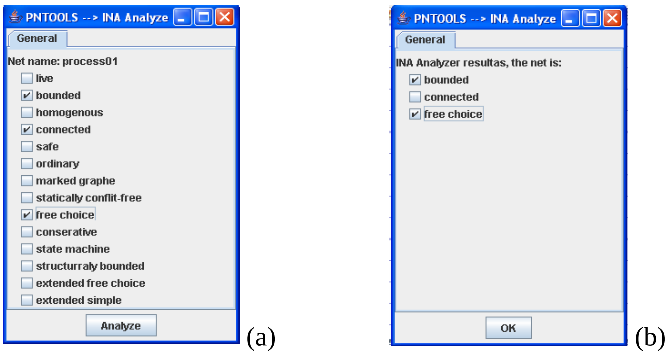


Fig.15. The net representation in **PNTools**.

4.2 INA analysis

After drawing a net, analyzing it in INA; comes to run INA/Analyze from menu bar. A window appears and looks like as shown in figure 16 (a).

We can select the properties to be analyzed by choosing analyze button. Then the results are obtained as presented in figure 16 (b).



Net name: process01
- □ live
- ☑ bounded
- □ homogenous
- ☑ connected
- □ safe
- □ ordinary
- □ marked graphe
- □ statically conflit-free
- ☑ free choice
- □ conservative
- □ state machine
- □ structurraly bounded
- □ extended free choice
- □ extended simple

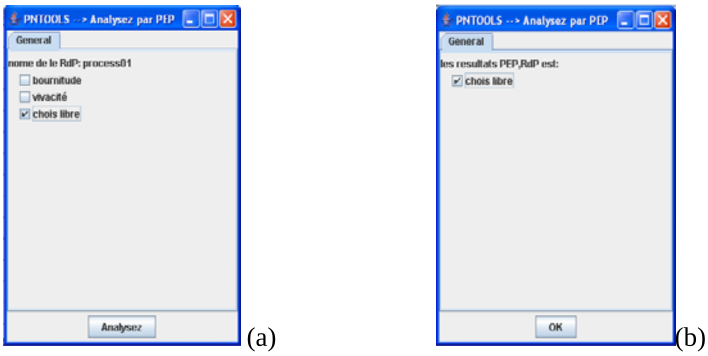
INA Analyzer resultas, the net is:
- ☑ bounded
- □ connected
- ☑ free choice

Fig.16. INA Analysis and results.

4.3 PEP analysis

After drawing a net, analyzing it in PEP; comes to run PEP/Analyze from menu bar. A window appears and looks like as shown in figure 17 (a).

We can select properties to be analyzed by choosing “analyze button”. Then the results are obtained as presented in figure 17 (b) (French interface).



nome de le RdP: process01
- □ bournitude
- □ vivacité
- ☑ chois libre

les resultats PEP,RdP est:
- ☑ chois libre

Fig.17. PEP Analysis and results.

5 Conclusion

This paper proposed a new multi-language environment to integrate Petri net tools. PNTOOLS is a contribution in the world of Petri net tools. We can say that PNTOOLS isn't only an environment to integrate existed tools, but it can also be a base to generate other new tools.

PNTOOLS recovers the problem of language interface and come with multi-language interface. The interface is detailed in this paper. In PNTOOLS, we have successfully integrated two of the famous Petri nets tools: INA and PEP. We believe that this integration open a new horizon in the Petri nets integration. In a future work, we plan to integrate other Petri nets tools and generate new tools.

6 References

1 Tadao Murata, “Petri Nets: Properties, Analysis and Applications”, Proceedings of the IEEE, VOL. 77, NO. 4, April 1989.

2 Matthias Jungel, Ekkart Kindler, and Michael Weber, “Towards a Generic Interchange Format for Petri Nets - Position Paper-”, XML/SGML based Interchange Formats for Petri Nets, Aarhus, Denmark, June 2000.

3 Adilson Arcoverde Jr., Gabriel Alve Jr., and Ricardo Lima, “Petri Nets tools integration through Eclipse”, University of Pernambuco, Department de Computer Systems, 2005.

4 Ekkart Kindler and Michael Weber , “The Petri Net Kernel: An Infrastructure for Building Petri Net Tools”, Humbolt Universitat zu Berlin – Institut fur Informatik, 1999.

5 Ekkart Kindler and Michael Weber, “The Petri Net Kernel: Documentation of the Application Interface”, Humbolt Universitat zu Berlin – Institut fur Informatik, 1998.

6 Ekkart Kindler and Michael Weber, “The Petri Net Kernel”, In: K. H. Mortensen, (ed.): Tool Demonstrations, Århus, Denmark, and 21st ICATPN. Jun. 2000, (pp. 71-75);

7 Ekkart Kindler and Michael Weber, “The Petri Net Kernel: An INA-Pilot”, Humbolt Universitat zu Berlin – Institut fur Informati, 1999.

8 Adilson Arcoverde, Jr., Gabriel Alves Jr., Ricardo Lima, Paulo Maciel, Meuse Oliveira Jr, and Raimundo Barreto, “EZPetri: A Petri net interchange framework for Eclipse based on PNML”, Centro de Informatica Universidade Federal de Pernambuco Recife, PE, Brazil, 2004.

9 Jonathan Billington and al, ” The Petri Net Markup Language: Concepts, Technology, and Tools”. Humbolt Universitat zu Berlin – Computer Science Department, 2003.

10 Bernd Grahlmann, ” Petri Net File Formats”. Universit at Hildesheim, Institut fur Informatik.

11 P. Starke and S. Roch. ”INA - Integrated Net Analyzer - Version 2.2”. Humbolt University of Berlin – Informatics Institute, 1999.

12 Elliotte Rusty Harold, ” Processing XML with Java”. http//:www.cafeconleche.org/books/xmljava.htm, 2001.

13 IGN / ENSG / CPRI, ” Programmer en Java”. IGN / ENSG / CPRI, France, 2002.

14 Raida El Mansouri, Elhillali Kerkouche, and Allaoua Chaoui, ” A Graphical Environment for Petri Nets INA Tool Based on Meta-Modelling and GraphGrammars”. Proceedings of world academy of science, engineering and technology, VOLUME 34, OCTOBER 2008.

15 Bruce Eckel, ” Penser en Java”. www.developpez.com,2003.

16 Eike Best, ”PEP Documentation and User Guide Version 1.8”. University of Oldenburg,1998.

17 Bernd Grahlmann,” The State of PEP”. University of Oldenburg, 1998.

18 B. Best and B. Grahlmann. “Pep - more than a Petri net tool”. In LCNS, volume 1055, pringer-Verlag, 1996 pages 397–401.

19 http://adiguba.developpez.com/librairies/shell/javadoc/, 2004

20 Vijay Gehlot, and Elliot B. Sloane. “Application of the Petri Net to Simulate and Validate Complex, Multi-Vendor, Heterogeneous Networks of Wireless Medical Device Systems”. Proceedings of the Tenth Americas Conference on Information Systems, New York, New York, August 2004.

21 Karsten Schmidt," LoLA : a Low Level Petri net Analyzer", Humbolt Universitat zu Berlin – Institut fur Informatik, September, 2000.

22 J. des Rivières and J. Wiegand, " Eclipse: A platform for integrating development tools", IBM SYSTEMS JOURNAL, VOL 43, NO 2, 2004.

23 Zhihui Yang, Wayne Zage, and Dolores Zage," The Eclipse Platform for Tool Integration and Development ", Ball State University, CS Department, Muncie, IN 47306, 2005.