Natural user interface integrated in Petri Net educating environment

Katarína Jelemenská*a, Peter Koinea, Ján Turoňa, Pavel Čičákā

*aFaculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, Bratislava 842 16, Slovakia

Abstract

This contribution deals with an educating tool called PN Designer, developed to support Petri Nets education - design, simulation, properties analysis and testing. It helps students to master Petri Nets and can also be used to examine students’ knowledge. Teachers are able to prepare various tasks that can then be used in the learning and/or assessment process. To support the assessment process the server part was designed to provide test creation, distribution, and automated evaluation facilities. Although primarily devoted to Tablet PCs, with some loss of comfort the environment can be used on traditional computers as well. The results show a great deal of the attractiveness of this kind of learning, and substantial time saving in the assessment process while preserving the assessment quality.

© 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.

Keywords: Petri Net; natural user interface; TabletPC; learning; assessment.

1. Introduction

Petri Nets belong to mathematical modeling tools suitable especially for describing discrete, distributed and parallel systems behavior. The application area of Petri Nets is very diverse including behavioral specification in the area of digital systems (Jurikovič et al., 2010), communication protocols (Adameit et al., 2010; Čičák et al., 2006), operating systems (Li et al., 2006), parallel programming (Tan et al., 2005), distributed database systems, production control (Gottschalk et al., 2008; Yasuda, 2010), etc. It is therefore natural that this topic be included in informatics and computer science study programs and courses all over the world (Kubátová, 2010).

Despite the wide range of applications supporting Petri Net (PN) design, simulation, and analysis there are still some issues that have not been addressed properly: automated skills based assessment and natural user interface in PN design. These two aspects are particularly relevant to the education process; therefore the main aim of our work was to allow the user to design the PN in the most natural way - drawing it as though on a piece of paper and integrating this interface with a multi-purpose environment supporting both learning and assessment processes.

When designing the functions of our educating tool we studied and tried many software and web applications. The most impressive were Netlab (Netlab, 2006) and Patrice Torguet’s Petri Network Simulator (Torguet, 1996). Netlab is characterized by its nice layout, simple control and highly sophisticated functions. The cooperation with another program, Matlab, is perhaps the most interesting quality which will draw your attention to this application immediately. It is fitted with basic functions like graphic editor, evaluation of properties including marking...
reachability, marking coverage, liveness, deadlocks, consistency as well as boundedness. The Petri Net simulation is implemented on an interactive basis. Users can fire individual transitions, which makes them able to model their own sequences of steps. These were the reasons why Netlab was the inspiration for our educating tool design and implementation. Patrice Torguet’s Petri Network Simulator is an old web application. Its user interface is uncomfortable, the number of its functions is very limited, and they offer very few and outdated functionalities. While Netlab inspired us mostly with its good features, (Torguet, 1996) was for us a warning about what we should avoid while designing our own application.

Although there are many applications, better or worse, dedicated to Petri Nets e.g. (TGI group, 2005; Flochová et al., 2006; Hudák et al., 2008; IBE, 2008; Mäkelä, 2002; van Hee et al., 2006) that provide the environment for PN design and simulation, most of them offering also the PN properties analysis, none of the known applications is directly suitable for testing the student’s knowledge and practical skills in designing Petri Net. This was the main motivation for designing a new tool that will provide the facilities like test creation, test evaluation, or shape recognition editor.

2. PN Designer features

The analysis of existing applications dedicated to Petri Nets helped us to specify the main requirements for our own tool. Simple, comfortable and user friendly interface is essential for our design. Shape recognition while drawing a Petri Net on a touch-screen with a stylus, ability to bend the edges, an easy way of changing object properties or deleting them are among the features that should also be included. The tool should be controlled mostly by a pen or mouse, without the need to use a keyboard unless it is really necessary. The main attribute of our solution is that it is a client-server application capable of creating and evaluating tests. That means it is intended to be used not only for education purposes but also for the evaluation of gained knowledge and practical skills in the area of Petri Nets, providing fast test evaluation as well as sufficient security. What is more, in most situations no Internet connection is needed.

2.1. Learning mode

Students have the possibility to learn by drawing their own Petri Nets, simulating them, and analysing their properties. PN Designer has a built-in real-time correction mechanism (disabled during an examination), which informs students of invalid actions (e.g. connection between two places). The students can even create their own questions and tests, using various types of questions. These tests can then be evaluated and thus the students can test their knowledge themselves. It also allows working on the pre-defined Petri Nets to help students understand some special problems concerning Petri Nets. However, the application supports just the traditional Petri Nets; the support for advanced Petri Nets, like colored, is not implemented. All the learning mode features are implemented on the client side of the tool. Therefore it can be used without the presence of a server or Internet connection.

2.2. Testing mode

Teachers can choose to give students different types of tasks to perform. Tasks vary from drawing a Petri Net with selected attributes, completing a pre-drawn Petri Net, redrawing an incorrect Petri Net, etc. As well as the primary function of this educating tool, which is drawing Petri Nets, teachers can use traditional on-line test questions, including multiple choice, single and multi answer, and matching questions. This gives the teacher the advantage of using a single test, created in one environment, to evaluate students’ theoretical knowledge as well as practical skills in the Petri Nets area.

A server is required for students’ examination. Its functionality is very limited since most features are implemented in the client part. The server just handles the user database, creates tests, encrypts or decrypts tests, and sends or receives the data. Because of this, the server hardware requirements are quite low. It does not even handle the test evaluation, which means the problem of the server overloading is avoided.
In every application which is intended for examining students’ knowledge, security is an important issue. The prevention of students cheating, securing objectivity and fairness of the exam and confidentiality of test questions are crucial requirements. We use an effective encrypting algorithm to encode data sent from the server to clients and vice versa, which ensures that the communication cannot be manipulated without notice. What is more, the educating tool generates a unique set of questions for each student. The examination test is stored for further analysis in case the students have some questions about it later.

2.3. Computer platform

PN Designer is designed to be as easy to use and comfortable as possible. That is why it was primarily intended to be used on the Tablet PC, providing touchscreen - the natural user interface. In spite of that, the regular PC can also be used, in the same way as it is with other known applications. Our carefully designed user interface makes all its functions available without touch-screen and stylus. Users are also allowed to use any kind of device that supplements the functionality of a computer mouse.

3. PN Designer architecture

The educating tool can be described as an autonomous client that handles all the mentioned innovations. That is the reason why this paper mostly refers to the client part. Optionally it can operate under the control of a light server. Both client and server can be divided into several components as shown in Figure 1.

To avoid test cheating, communication between client and light server is encrypted on both sides by a Scrambler module based on a block encryption method with a user specific key. Locally stored data is manipulated by standard MFC (Microsoft Foundation Class Library) interface functions (Microsoft, 2010). The Draw recognition module is added in order to enable more intuitive and easier interaction. To avoid flickering and to improve performance we decided to replace standard MFC drawing functions with the specially designed Presentation component, based on Open GL (Open Graphics Library) (Khronos Group, 2010), which provides fast and modern interface without
flickering of the screen, which is caused by redrawing. The algorithms designed to evaluate Petri Net parameters and properties and to compare them with correct answers are implemented in the Correction entity.

As well as the Scrambler, the light server consists of the Authentication module and Test generator. Both of these use three databases: User DB, Tests DB and Questions DB. The Test generator is used to create various tests, selecting questions from the questions database randomly.

4. Interaction and user interface

Interaction is a crucial part of any educating tool. In our case the application was primarily developed for the Tablet PC, therefore, there was an urgent need to design an appropriate interface. The user interface is given in Figure 2. There are two different types of main document. If we select the New Document in the menu toolbar, the system creates a new test with one free drawing question. Here the students can draw their own Petri Nets, and simulate and analyse them. The different test type is created selecting the New Test field in the File menu. This enables the adding of more questions using the Add question function.

Our aim was to enable the use of a stylus as an ordinary pencil. This means users can sketch a circle-like shape on the screen and the system will recognize it as a place and replace it with a neat circle (See Figure 3). To draw a transition, a straight line should be sketched. Again the system recognizes it and draws the right image. To draw an edge the user starts drawing a line from one object (place or transition) to the other. The edge is then created exactly as the user drew it. The criss-cross gesture is recognized as a delete command. The objects can be selected with a pen and moved anywhere on the screen, or their properties can be edited. The interaction is very user friendly and made as simple as possible, giving the students the feeling that they are actually sketching on a sheet of paper, rather than on a computer. To achieve this aim stylus and touch-screen technology, Pen API, and Ink API methods have

Figure 2. PN Designer user interface

Figure 3. Shape recognition function.
Sketching with a traditional mouse is far from comfortable. Therefore we implemented menu bars and tool bars with corresponding actions as an alternative. Clicking for example on a transition button activates the transition mode, and then with each subsequent click on the canvas a new transition is created.

5. Results and experience

PN Designer was implemented using an object-oriented approach and C++ programming language in Microsoft Visual Studio 2008 SP1 under the operating system Windows 7 Professional. In academic year 2010/2011 the pilot run took place at the Digital systems description course, taught at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. Altogether, 60 students were involved, using the tool for education as well as testing purposes, thus replacing the previously used pen and paper exam. The research was conducted on student attitudes and learning outcomes. The collected data were analyzed and the selected results are given in Figure 4. Average test attainment comparison in two consecutive academic years shows the evident improvement in students’ results. The questionnaire inquiring about student attitudes was filled out by 43 students. The students appreciated especially the fact that they can do the exam in the environment they are already familiar with from the learning process.

6. Conclusion

The PN educating environment, designed and implemented as a result of this work, enables students to understand basic principles of PN design and simulation and to gain basic designing skills by means of solving the integrated exercises. Teachers can prepare various tasks that can then be used in the learning and/or assessment process. To support the assessment process the server part was designed to provide test creation, distribution, and automated evaluation facilities. The PN Designer’s natural user interface is simple, very intuitive and the vast majority of students found it very comfortable. However, the educating tool not only provides the enhanced Petri Net editor with the shape recognition function, but also supports Petri Net simulation and properties analysis. Naturally, with some loss of comfort it can be used on traditional computers as well. The results show a great deal of attractiveness for this kind of learning and substantial time saving in the assessment process while preserving the assessment quality.

Although there are many applications supporting Petri Net design, simulation, and properties analysis, based on the authors’ knowledge, none of the known applications provides facilities like test creation, test evaluation, or shape recognition editor. This is the main enhancement that was brought by our educating tool.

Acknowledgment

Support from the Slovak Science Grant Agency (VEGA 1/0649/09 “Security and reliability in distributed computer systems and mobile computer networks”) and HP Technology for Teaching grant 2007 “Mobile Education Center” is gratefully acknowledged.
References

Adameit, S. et al. (2010). Modelling distributed network security in a Petri net- and agent-based approach. Multiagent System Technologies. Lecture in Computer Science, 6251/2010, 209-220.

Čičák, P., & Mostafa, H. E. (2006). Hands on roaming duration: Petri-Nets modelling of a wireless mobile-IP procedure in Cisco platform. Proceedings of International conference on Networking and Services (ICNS'06) (pp. 28-33). Washington, DC, USA: IEEE Computer Society. doi: 10.1109/ICNS.2006.55

Flochová, J., Auxt, F., Radakovˇič, M., & Jombik, O. (2006). PNDesigner a tool designed for model based diagnosis and supervisory control of DES. Proceeding of WODES - Workshop on discrete event systems (pp. 471-472). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/WODES.2006.382521

Gottschalk, F., van der Aalst, W. M., Jansen-Vullers, M. H., & Verbeek, H. M. (2008). Protos2CPN: using colored Petri Nets for configuring and testing business processes. Int. Journal on Software Tools for Technology Transfer (STTT), 10 (1), 95-110.

Hudá, S., Koreˇcko, Š., & Šimoňák, S. (2008). A support tool for the reachability and other Petri Nets related problems and formal design and analysis of discrete systems. Problems in programming, 20 (2-3), 613-621.

IBE Simulation Engineering GmbH (2008). PACE 2008. Retrieved May 10th, 2011, from http://www.ibepspace.com/Eng/what_is_pace.html

Jurkoviˇc, M., Čičák, P., & Jelémenská, K. (2010). Parallel controller design and synthesis. Proceedings of the 7th FPGAworld Conference (pp. 35-40). New York, NY, USA: ACM. doi: 10.1145/1975482.1975486

Khranos Group (2010). OpenGL API documentation overview. Retrieved May 25th, 2011, from http://www.opengl.org/documentation/

Kubátová, H. (2010). Teaching principles of Petri Nets in hardware courses and students projects. In E. M. Abu-Taieh, & A. A. El-Sheikh (Eds.), Handbook of Research on Discrete Event Simulation Environments: Technologies and Applications (pp. 178-190). IGI Global. doi:10.4018/978-1-60566-774-4.ch008

Li, Z., & Wie, N. (2006). Deadlock control of flexible manufacturing systems via invariant–controlled elementary siphons of Petri Nets. The Int. Journal of Advanced Manufacturing Technology, 33 (1-2), 24-35.

Mäkelä, M. (2002). Maria: modular reachability analyser for algebraic system nets. In J. Esparza & Ch. Lakos (Eds.), Application and Theory of Petri Nets 2002, ICATPN 2002, volume 2360 of Lecture Notes in Computer Science (pp. 434–444). Berlin: Springer-Verlag.

Microsoft (2010). MSDN Library. Retrieved May 25th, 2011, from http://msdn.microsoft.com/en-us/library/default.aspx

Netlab (2006). A tool for modelling and analysis of Place/Transition (P/T) Petri Nets. Institute of Automatic Control, RWTH Aachen University. Retrieved May 25th, 2011, from http://www.irt.rwth-aachen.de/fileadmin/IRT/Download/NetLab/HilfeHTML/

Tan, R., & Guan, S. (2005). A dynamic Petri Net model for iterative and interactive distributed multimedia presentation. IEEE Transactions on Multimedia 7(5), 869–879.

TGI group (2005). Petri Nets world site. University of Hamburg, Germany. Retrieved May 25th, 2011, from http://www.informatik.uni-hamburg.de/TGI/PetriNets/

Torguet, P. (1996). Petri Network Simulator v1.0. Java Applet. Paul Sabatier University in Toulouse. Retrieved May 25th, 2011 from http://torguet.free.fr/java/Petri.html

van Hee, K., Ounea, O., Post, R., Somers, L., & van der Werf J.M. (2006). Yasper: a tool for workflow modeling and analysis. Proceeding of The Sixth International Conference on Application of Concurrency to System Design (ACSD’06) (pp. 279-282). Los Alamitos, CA, USA: IEEE Computer Society.

Yasuda, G. (2010). Distributed cooperative control of industrial robotic systems using Petri Net based multitask processing. In H. Liu et al. (Eds.) Intelligent Robotics and Applications. Lect. Notes in Comp. Science, Vol. 6425/2010, 32-43. Berlin Heidelberg : Springer-Verlag.