PETRI NET EDUCATING TOOL

Ján Turoň, Šimon Hupka, Michal Choleva, Katarína Jelemenská, Peter Koine, Emil Močovský
Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies,
Slovak University of Technology, Ilkovčová 3, 842 16 Bratislava, Slovakia
e-mail: turon_jan@zmail.sk, jelemenska@fiit.stuba.sk

Abstract. This contribution deals with an educating tool, designed to support Petri Nets education -
design, simulation, properties analysis and testing. It will help students to master Petri Nets and will
also be used to examine the student's knowledge. It automatically evaluates the student's exam
work, reducing thus the time of exam evaluation from days to seconds. The educating tool is primary
developed to be used on Tablet PC computers, so that students have the feeling they are drawing on
a sheet of paper giving them a much more comfortable environment. However, it can also be used on
traditional computers without substantial loss of comfort. The designed architecture is object
oriented and divided into client and server part, where the client has most of the functions, while the
server is only used on exams. The tests will be taken on client computers, not directly on server.
Therefore there will not be a situation where all the solved tests would have been lost due to server
malfunction.

Keywords: education, e-learning, skills assessment, learning object, software, training, Petri Nets

1. INTRODUCTION

Petri Net is one of the several mathematical modelling tools that is suitable especially for describing the behaviour of
discrete, distributed and parallel system. It is a graphical and mathematical tool with two kinds of nodes - places and
transitions as well as directed edges that can interconnect two nodes of a different kind. The state of Petri Net
execution is represented by so called marking. The application area of Petri Nets is very diverse. For instance
they can be used in product lines to automate and maximize production or to develop the behavioural specification of a
digital system.

When designing the functions of our educating tool we studied and tried many software and web applications. The
most impressive were Netlab and Patrice Torguet’s Petri Net Network Simulator.

Netlab [2] 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 interactive level. Users can fire individual transitions, which makes them able to model
their own sequence of steps. These were the reasons, why Netlab was the inspiration for our educating tool design and
implementation.

Patrice Torguet’s Petri Network Simulator [1] is a web application. Its working interface is uncomfortable, the
number of its functions is very limited, and they offer very few and archaic functionality. While Netlab inspired us
mostly with its good features, this web application was for us a warning, what should we avoid while designing our
own application. On the other hand, this application is quite old, so the worse quality understandable.

Although there are many applications, better or worse, dedicated to Petri Nets (e.g. [3][4][5][6][7]) that provide the
environment for Petri Net design and simulation, most of
them offering also the Petri Net 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 to design a new
application, that will provide us with facilities like test
creation, test evaluation, or shape recognition editor.

2. EDUCATING TOOL FEATURES

SPECIFICATION

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 touch-screen with a stylus, bending the edges,
easy way of changing object properties, or deleting them
are among the features, that will not be missing in our
application. We want our tool to be controlled mostly by a
pen or mouse, without the need to use keyboard, unless it is
really necessary. The main attribute of our solution is that it
is 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 evaluation of
gained knowledge and practical skills in the area of Petri
Nets.

The educating tool is programmed to be as easy to use and
comfortable as possible. It is intended to be used primary on
Tablet PC, but can also be used on regular PC just like other
known applications. It can be used anywhere, whether there
is Internet connection or not. What is more, it provides fast test evaluation as well as sufficient security.

Scope of usage
The educating tool should be primarily used on Tablet PC as mentioned before. But not everyone has Tablet PC at his disposal. This is not a problem of course because one can use our educating tool on regular PC as well. 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 supplementing the functionality of a computer mouse.

Server and Internet independence
Most of the features are implemented on the client side of an educating tool. Therefore the users can use it without the presence of a server for the purpose of learning how Petri Nets work. It also allows working on the pre-defined Petri Nets to help students understand some special problems concerning the Petri Nets. The server is required only for student’s examination. It is not required to have Internet connection while using the application.

Testing security
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 the crucial requirements. We use effective encrypting algorithm to encode data sent from server to clients and vice versa, which ensures that the communication cannot be manipulated without notice. What is more, the educating tool generates 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.

Tests questions variability
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. Despite the primary function of this educating tool, which is drawing the Petri Nets, teacher can use traditional on-line test questions, including multiple choice single and multi answer and

---

![Application architecture](image)

**Fig. 1.** Application architecture.
matching questions, as well. This gives a teacher the advantage of using single test, created in one environment, to evaluate student’s theoretical knowledge as well as practical skills in the area of Petri Nets.

Practice – an attractive way of learning
Students have the possibility to learn by drawing their own Petri Nets, simulating them, and analysing their properties. Our educating tool has a build-in, real-time correction mechanism (disabled during the examination), which informs students of invalid actions (i.e. 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. However, the application supports just the traditional Petri Nets, the support for advanced Petri Nets, like coloured, is not implemented.

3. ARCHITECTURE

Our 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 client part. Optionally it can operate under control of 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 using user specific key. Locally stored data is manipulated by standard MFC (Microsoft Foundation Class Library) interface functions [16]. The Draw recognition module is added in order to enable more intuitive and easier interaction. To avoid flickering and better performance we decided to replace standard MFC drawing functions by specially designed Presentation component based on Open GL (Open Graphics Library) [4]. Usage of Open Graphics Library provides fast and modern interface without flickering of screen caused by redrawing.

The last component is Correction entity. The algorithms designed to evaluate Petri Net parameters and properties and to compare them with correct answers are implemented here.

Except for scrambler, the light server consists of an Authentication module and Test generator. Both of these use three databases: User DB, Tests DB and Questions DB. Naturally, the Test generator is used to create various tests, selecting questions from questions database on random basis.

The provided architecture refers only to the most necessary components.

![Fig. 2. Application user interface example](image_url)
4. INTERACTION and USER INTERFACE

Interaction is a crucial part of any educating tool. We were developing the application that will primarily be used on Tablet PC computers. Therefore, there was an urgent need to have an appropriate interface.

The user interface is given on Figure 2. There are two different types of the main document. If we select the New Document in menu toolbar, the system creates a new test with one free drawing question. Here the students can draw their own Petri Nets, simulate and analyse them. The second document will be created after selecting the New Test field in File menu option. One can add more questions to this test later, simply by using the Add question function.

We wanted to enable the use of Tablet PC stylus as an ordinary pencil. This means users can draw a circle or a similar shape on the screen and the system will recognize it as a place and draw a proper image on the position where the actual drawing started (See Figure 3). The place can be then selected with a pen and moved anywhere on the screen or we can change its properties. To draw a transition, user draws a line. 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 another object. The edge is then recreated exactly as the user drew it. To delete an object, just draw a cross over the object and it will be deleted. To delete more objects at once, the user must draw one big cross over the objects he or she wants to remove.

As mentioned before, the interaction is very user friendly and made as simple as possible. It should give students the feeling that they are actually drawing on a sheet of paper, rather on a computer.

However, not everybody has a Tablet PC. Therefore we implemented menu bars and tool bars with corresponding actions as an alternative. By clicking for example on a button named transition the transition mode is activated and we can then place transitions on canvas right where we click. But, if the user wants, he or she can still use the mouse to draw circles or lines even though it’s less comfortable.

5. DATA STRUCTURES

In order to store and manipulate application’s data, we created four base classes. A test is represented by class CPNTest, a question by CPNQuestion and a Petri Net is represented by CPNPetriNet class (See Figure 4). A test usually comprises several questions, some of which may contain Petri Net information. It all depends on the question type.

Petri Net, as a matter of fact, is a new question type, where students are expected to draw a new Petri Net or finish an existing one to meet the given specification. However, text based questions with multiple choices do not need any Petri Net information at all. Optionally, it can be included as a read only illustration.

![Fig. 3. Shape recognition function](image)

![Fig. 4. Relationship and inheritance among classes](image)
Petri Net is made up of places, transitions and edges. Places and transitions have been implemented in classes called CPNPlace and CPNTransition. Place, naturally, is a circle with user specified radius. Transition is shown as rectangle. Each transition and place has a label. To be able to display and change the position of this label, we created a class called CPNLabel. This means, that a transition or place label is an object with its own x and y coordinates, as well as other parameters. There is no object to represent the edges. Edge is supposed to be a link between two or more objects. Said in C++ jargon, each object contains pointer to the next object on the edge. In general there are no limitations, but for proper Petri Nets source and destination object they must be of a different kind. Additionally the nodes may be inserted between them. Node, as CPNNode class is designated to shape these edges. Otherwise edges will only be straight lines between places and transitions. The fourth base type is CPNObject representing a generic Petri Net object. The purpose of this abstraction is to unify operations with places, transitions, nodes and labels. It also makes it easier and faster to develop and improve the application. Generic objects contain common parameters such as unique identifier, position and a list of linked objects.

6. CONCLUSIONS

In this paper an educating tool was presented, which should help students to learn and improve their skills regarding the Petri Nets domain. It is also the application that will be used to examine students’ theoretical knowledge and practical skills with an advantage of automatic correction and evaluation of their exams in a matter of seconds. Therefore the application can run in two modes, the free drawing mode where users can draw Petri Nets and try different functions. Users can create their own tests, share the tests with other users and let the application evaluate them. What is more they can draw a Petri Net, simulate it to see how the designed model works and let the system find out its properties. The second mode, test mode, will be used during examinations. The automatic evaluation function reduces the demanding and time consuming work of teacher spent on correcting and evaluating pen and paper exams. The educating tool is designed to have a simple interface so the students have a feeling they are actually working with a sheet of paper. Users draw symbols on the screen, the system recognizes them as shapes and eventually creates the correct objects. The designed educating tool provides not only the enhanced Petri Net editor with shape recognition function, but supports also simulation of Petri Nets, as well as Petri Net properties analysis. On the other hand all the functions support only traditional type of Petri Net. Although there are many applications supporting Petri Net design, simulation, and properties analysis, 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. The educating tool is planned to be used for education as well as testing purpose in the course Digital system design at our faculty. It will replace the previously used pen and paper exams. We believe that the students will appreciate especially the fact that they can do the exam in the environment they are already familiar with from previous learning process.

7. REFERENCES

[1] Torguet, P.: Petri’s Petri Net Java Applet, Universität Hamburg, MIN Faculty, Department of Informatics, [Last access February 20th, 2009]. Available at: http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/java/Patrice/


Khronos Group.: OpenGL API Documentation Overview. [Last access February 20th, 2009]. Available at: http://www.opengl.org/documentation/


THE AUTHORS


ACKNOWLEDGEMENT

This work was partially supported by 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”.

This page contains information about various scientific works and publications, including contributions by different authors on topics such as Petri Nets, supervisor control, and algorithms in Java. It also mentions the utilization of OpenGL API documentation and Microsoft MSDN Library. The page highlights the work of several students and researchers, including Bc. Ján Turoň, Bc. Michal Choleva, Bc. Šimon Hupka, and Bc. Emil Moťovský, who have contributed to projects in the fields of computer systems and networks, as well as other scientific interests. The page includes references to publications, conferences, and awards received for their work.