TEACHING TIMED HYBRID PETRI NETS FOR PERFORMANCE MODELING OF COMPUTERS NETWORKS PROTOCOLS

Emilian Guţuleac, Victor Şontea

Technical University of Moldova, Computer Science Department,

Bd. Ştefan cel Mare, nr. 168, MD-2012, Chişinău, Republic of Moldova
e-mail: {egutuleac, sontea}@mail.utm.md

INTRODUCTION

The handling of abstraction and of models is generally claimed to be one of the central issues of computer engineering and science. However, little attention is paid to the process of model creation, validation and performance evaluation of computer networks protocols.

Inside Performance Modeling of Computer Systems course, processes synchronization and/or communication is one of the most important and difficult issues to teach and to understand. Related to this hard problem we can find some basic problems as deadlock avoidance, fairness, conflicts, etc. Mentioned course prerequisites lack of minimal understanding and skills related to these basic concepts or concurrent processes. In this context, we need some formal tool which can help to: a) better understand basic concurrency concepts; b) easily visualize different synchronization/communication schemes; c) straightforward translation the performance measure into some target visual tool; d) use automated or semi-automated software tools to explore these schemes properties of models.

In this area more main formalisms have been discussed in the literature, namely Petri nets (PN) and Timed Hybrid Petri Nets (THPN) [1]. THPN are mathematical formalism intended to be used for modeling, simulation and performance analysis of different kinds of systems. In computer science THPN are used for modeling a great number of either hardware and software systems, or various applications in computer networks. A special advantage of THPN is their graphical notation, which reduces THPN model learning time and simplifies their use. Hence PN are used for teaching numerous concepts in computer science [2]. They can greatly help us in the education of these topics. In the one hand, they give us an intuitive graphical representation behavior for processes of the network protocols.

VPNP TOOL

VPNP [1] is a window-based, object-oriented tool, in which elements typical of hybrid Petri net models (places, transitions, and arcs) are manipulated under the assistance of basic syntactical rules that prevent the construction of incorrect models. This tool is designed with a broad functionality: it contains a large number of various extensions to stochastic and timed hybrid Petri nets that have been proposed, user-defined place and transition rewards, place capacities, discrete and continuous arc marking-dependent multiplicities, firing marking-dependent firing discrete and flows rates, single and infinite-server transitions, and memory policies. The model drawn using the interface is internally translated into several files that are however completely transparent to the user. The representation of THPN model is thus the object to which all the command provided by the Graphical User Interface.

The workspace is divided into two parts: on the bottom side there is a navigation tree of THPN objects; on the left side, where is a document review of the THPN graph model which is being operated at the moment. Names of all transitions and places of the active THPN are listed in the right side of the window. Features of the selected object, such as name, transition guard function, place marking, can be changed. This tree has proven to be very useful in practice, since the student got accustomed to the similar user interface organization during their previous software engineering courses. It is also possible to write annotations and quantitative attributes for some objects of the graph shown on the right side of the window. The graph of the active THPN is drawn and the modeling function performs in this window. The results of the simulation as well as a single simulation step are also shown.

While places are represented as circles, transitions are in the form of rectangles that become two times wider after a transition is fired – this is useful for learning and better detection of fired transitions. Net nodes are linked by arcs, which were outlined curves. There is no limit in the number of breaking points. To link two nodes it is necessary to select (using the mouse) a starting node (place or transition), then the arc breaking points finally, a target node of the THPN. If a user tries to link nodes of the same type (for example place and place), such linking would be rejected and a dialog would appear giving the information about syntax error that has been made. This VPNP feature is useful in the process of learning the THPN. The firing history has been developed to present the sequence of parallel fired transitions during the execution of a specified number of simulation steps. In this way one can observe the simulation procedure and conflict situations. Consequently, this tool is convenient for realizing the need of THPN model analysis, and that is why students should use it in the first phase of learning THPN. A set of parallel fired transitions include information about simulation step that the system reached in certain state. An important factor that makes learning easier is interaction between analysis results and the deserved model.

SIMULATION AND PERFORMANCE EVALUATION OF TIME-OUT PROTOCOL

The procedure for simulation and evaluating performance of protocols with timed stochastic and hybrid Petri nets is the following: definition of hypothesis and assumptions; definition of performance measures; definition of studies, or measurement situations, trying to cover the main real and generic situations; simulation and evaluation of the measures for both strategies; comparison and deduction of its behavior in more generic situation.

The main hypothesis and assumptions are related to the type of traffic introduced in the network and to the capabilities of the models. For example, traffic by greedy sources in order to study worst case situations. Message losses and retransmissions are or not considered and all remote traffic is assigned high priority.

In this section, we present an example illustrating the application of our synthesis simulation approach. This example demonstrates the timeliness behavior of a protocol. A THPN model of a simple protocol with a time-out mechanism is shown in Fig.2. The token in p_1 represents a message to be send from a “sender” (p_1) to a “receiver” (p_3) an confirm by an acknowledgement (p_7) sent back to the sender. The message is sent by firing t_1, after which a single token is deposited in p_2 (the message) and in p_4 (the reset of copy “retransmitted” mechanism) and in p_9 (the reset of time-out). Enabled t_2 and t_{10} can start their firings concurrently; firing time of t_2 represents the “communication delay” of sending a message, and that of t_8, the time-out time. When the firing of t_2 is finished, a token is deposited in p_5, the receiver. p_2 is a free-choice place, so t_2 and t_{10} are enabled simultaneously, but only one of them can fire (inhibitor arc (p_{12}, t_2) enable or disable t_7). Transition t_{10} represents (is a simplified way) the loss of distortion of the message or its acknowledgement; it t_{10} is selected for firing, the token is removed from p_{11} as well as from the model. In such a case the time-out transition t_8 finishes its firing with removed token in p_{10}, so the arc (p_{10}, t_4) and (p_4, t_4) enables t_4, and its firing regenerates the lost token in p_2 and p_{16}, so the message can be “retransmitted”.

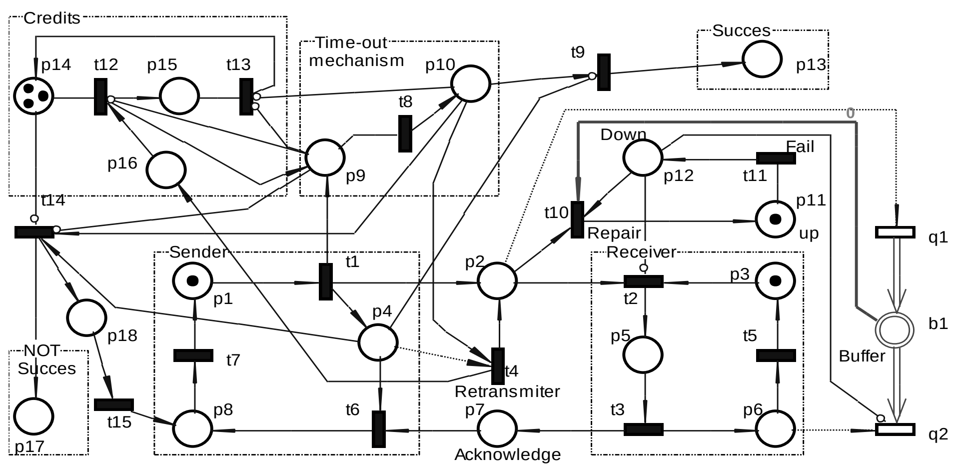

Fig.2. A simple protocol with a time-out.

If the message is received correctly, t_2 is selected for firing rather than t_{10}, and after another “communication delay” (modeled by t_2 and t_3) tokens are deposited in p_6 and p_7 (so another message can be sent to the receiver). The token in p_7 and in p_4 waits until the received acknowledgement transition t_8 finishes its firing, and than removes the token by firing t_6 (in the loss of distortion of the message or its acknowledgement the t_9 is disabled in this case by the inhibitor arc). The firing times of timed transitions are selected in such a way that the time-out time (t_8) is greater than the sum of the delays of sending a message (t_2) and acknowledgement (t_3 and t_6).

The continuous process of deposit and remove of bytes in buffer is modeled respectively by the continuous timed transitions q_1 and q_2 and continuous place b_1. The jump arc (b_1, t_{10}) connected continuous place b_1 and discrete timed transition t_{10} (the distortion of the message) describe the capability of a transition to empty in zero time the existing fluid level from a continuous place b_1 then it fire.

CONCLUSIONS

The THPN formalism has been used in different application fields. In some cases it has been used to overcome the state space explosion problem. This paper presents a case study where the THPN formalism is used to simulation of protocol which VPNP. The emphasis of the present investigation is to show two interesting features of the ability to model "real" continuous quantities, and their mutual interaction with states of the discrete part of the system. VHPN calculation engine does not limit number of objects in a THPN structure, so it possible to use the program in modeling network protocols, computer hardware and even non-computer related areas.

REFERENCES:

[1] Guţuleac, E., Reilean, and A. Boşhneaga C. VPNP - software tool for modeling and performance evaluation using generalized stochastic Petri nets. In: Proc. of 6th International Conference on DAS2002, Suceava, România, 2002, pp. 243-248.

[2] Desel J. Teaching system modeling, simulation and validation. In Proceedings of an 32th Conference on winter simulation, 10-13 December, FL, USA, 2000, pp 1669-1673.