

VCPN TOOL FOR PERFORMANCE MODELLING OF COMPUTERS SYSTEMS

Emilian Guțuleac, Odobescu Diana, Emilia E. Guțuleac

Technical University of Moldova, Computer Science Department,

E-mail: egutuleac@mail.utm.md

INTRODUCTION

Petri nets (PN) are well established to support the modeling and simulation of parallel and concurrent systems and represent a well-known and widely used formalism. Coloured Petri Nets (CPN) have been introduced to allow the modeler to make more manageable descriptions of large PN models, by folding equal subnets into each other [1, 5]. This folding consists of representing all instances of a process type by a single subnet and distinguishing the individual processes of this type by different colours.

We present the Visual Coloured Petri Nets (VCPN) software tool (C++) for modeling and verification of computer systems using timed CPN. This tool that has been developed for these purposes is illustrated with an emphasis on the modelling environment it provides and on its software architecture. Future enhancements of this tool are also discussed.

VCNP TOOL OVERVIEW

VCPN is a software tool for construction, simulate and analysis untimed and timed CPNs.

This tool consists of a Graphical User Interface (GUI), an animation modeling and a visualization component. The GUI is designed with multi-document capability, allowing several nets to be edited simultaneously. This allows the user to study the behavior of the net by observing the token game. The GUI has no menu bars or pull-down menus, and only few scrollbars and dialog boxes.

We require colour sets to be finite in order to ensure that a coloured net may be “unfolded” into its equivalent flat PN. The set of possible colours of transitions is determined by the colour sets of its input and output places and the coloured arc functions between the transition and these places. Places can be assigned an coloured initial marking and a maximum capacity. Transitions have the largest numbers of parameters. Immediate transitions can be assigned a weight firing probability, whereas timed transitions can be assigned a firing exponential time distribution chosen from a large set of alternatives. VCPN also allows the use of marking-dependent guard functions and priorities to control the enabling of transitions, and also supports multiplicity of arcs, whose may be specified as a constant or with a function. We see all the places, transitions and arcs. Bold transition is a

timed transition. Immediate transition is a transition with zero firing time. Tokens of places are represented by colour set.

Coloured Reachability Graph (CRG) describe the state space of CPN, also it includes information about transitions, whose operation caused change of marking in the model and urged a new state and determine several important qualitative and logical behavioural properties of the model, such as: reachability, boundedness, liveness, reversibility and steadiness of model functioning [4].

Main toolbar of GUI contains all the necessary tools for designing, editing and analysing the topology of the CPN, the Descriptive Formula (DF) of PN [2, 3], scale tool and help button: a) *Arrow* – the purpose of this tool is to select and to move objects, to move selections and the changing object's attributes; b) *Designing* are used to design the CPN; c) *Editing* – are represented by standard of copy, paste and cut buttons and grid button – showing and hiding the grid; d) *MacroPlace* – draws a macroplace for hierarchical representation of CPN; e) *Reachability Graph* – invoke the new process thread to start qualitative analysis.

The thing that accentuates our application is tool for making descriptive formula (DF) of PN and CPN [2]. The DF is a sequence of symbols-places of the elementary formulas connected by compositional operators, so that the identical names of places are stick together. The DF of PN describes it only according to the topology of net. So it is bared of unnecessary thing that has the graphical representation such as physical coordinates of objects. There are 4 different kinds of elementary compositional operations [2, 3]:

- *sequence* (“|”) – when the places are sequential connected by means a transitions;
- *synchronisation* (*merge* “*”) – when a certain number of arcs which were started from several places have their endings at one certain transition;
- *distribution* (*split* “;”) - when a number of arcs which were started from certain transition have their endings at the several places;
- *competing parallelism* composition (“V”) of subnets described by elementary formulas where the places which same name (label) are fused.

In Figure 4 is represented an example of a DF relative to a specified PN.

Object Inspector provides easy and fast access to every CPN objects property. That while changing property of the selected object, VCPN automatically changes the values of all the items, which depend on this value, and changes the general net topology of the project. User can observe all the changes immediately by means of the hierarchical tree of Object Inspector. It is very useful in case user's changing the topology of the network modelled.

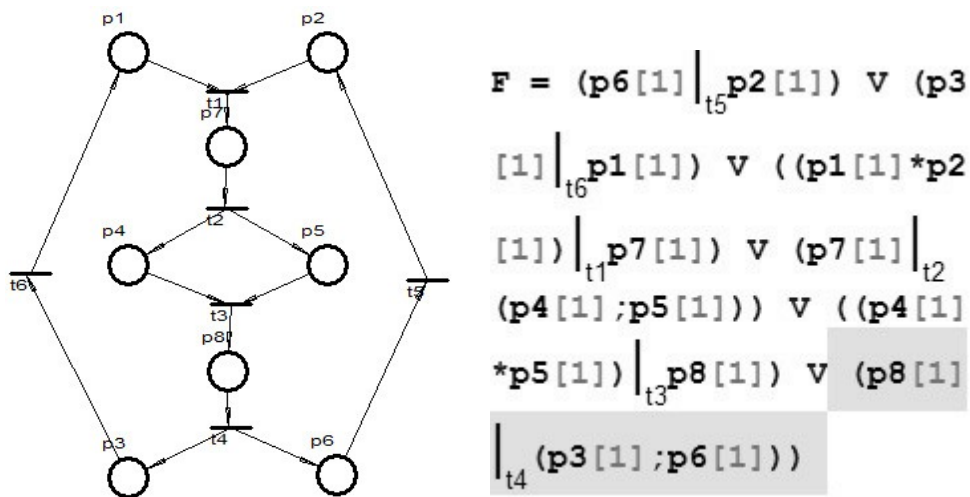


Fig. 1. An example of PN and its descriptive formula F.

File browser. To make it easier to work with files, the File Browser was included in VCPN application. This feature allows us to browse all the filenames of the projects in the Sample Directory, which makes switching between the projects more convenient.

The VCPN interface was made up as an user-friendly interface by means of standard GUI objects so the user will not waste his time searching for any tool to use.

The VCPN interface was made up as an user-friendly interface by means of standard GUI objects so the user will not waste his (or her) time searching for any tool to use. User can get access to any object at any time by a simple mouse click (screen shot for MP1 and for MP2 contain MP1). For example, to create a MacroPlace position user just has to click “New” button in the *Library window*. To delete this object, user just has to click the “Delete” button and to specify the object. views: collapsed (only macroplace symbol is reflected), expanded (all the relative macro places for the specified object are in the Library Window user also can edit the specified MacroPlace and even can place one MacroPlace position from the library into another MacroPlace position by a simple choosing the desired element in the main toolbar and clicking (see the screenshot for the MP1 and MP2 objects in the library). To place the created object into the workspace user can simply drag the specified object from library to the workspace. Let us notice, that a Macro object in the workspace can be viewed in 3 different ways (being viewed) and “expand all” – all the hierarchical structure of a MacroPlace position is being reflected in the workspace. In the library window user can also specify the *start object* and the *stop object* for a macroplace position.

The VCPN tool provides easy access to the most frequently used tools by providing a toolbar consisting of *ImageButtons* which can be depressed to enable functions. The tool also provides a collection of editing features including *Cut*, *Copy*, *Paste*, and *Delete* to make the design process easier. The view of the design can be altered to zoom-in and zoom-out on portions of the design.

By providing a time parameter for each transition, we are able to simulate the net performance. Currently, this time parameter can either be constant or exponentially distributed. In VCPN tool a timed CPN is simulated in two modes: auto and step by step. In the auto mode, transitions are fired as long as they are enabled. In step mode, transitions are fired according to the step chosen by the user. In all cases, token flow due to firing can be visualized on the screen.

Developing Environment: Microsoft Visual Studio 6.

CONCLUSION

VCPN is under construction software tool. It combines powerful engine together with user-friendly intuitive interface. This program does not require knowledge of all algorithms using for analysis. Timed CPN support in VCPN enables user to create CPN models describing of computer systems of any complexity.

Already VCPN is used in teaching process; it gives students a grasp of Petri net principles, introducing them into world of performance evaluation of computer systems. It is planned to add support of coloured hybrid Petri nets, structure analysis (P and T-invariants) and any other features that can possibly enlarge VCPN 's ranger of application or will make working with Petri nets simpler and faster.

REFERENCES

- [1] H. Alla, R. David. *De Grafctet aux reseaux de Petri*. Ed. Hermes, Paris, 1992
- [2] E. Guţuleac, *Modelarea și evaluarea performanţelor sistemelor de calcul prin reţele Petri Partea II*. Ed. U.T.M., Chişinău, 1999
- [3] E. Guţuleac, V. Moraru, R .Pogurschi. *Formules Descriptives de Reseaux de Petri et la Modelisation des Systems Distribues*. Intern. Conf. on D&AS, Nr.9, Suceava, România, May 1998, pp. 51-58, 1998.
- [4] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. *A symbolic reachability graph for coloured Petri net*“, *Theoretical Computer Science*, 176 (1-2), pp.39-65, 1997.
- [5] K. Jensen, *Coloured Petri nets*, Springer-Verlag, 1992.