EMASE : a Petri Net Tool

GUAT YEW TAN
School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang
email : gytan@cs.usm.my

Abstract

A new Petri Net object-oriented tool termed Extended Modeling Assisted Software Environment (EMASE) is proposed. It is planned to adopt hierarchical and object-oriented structure, and will provide a Graphical User Interface (GUI) environment for the creation, modification, analysis of the modeled system. The proposed package will allow the creation of objects (places, transitions, tokens and arcs), moving and erasing of objects, snapping and setting grids for alignment, creation of sub and super-nets. Further, automatic simulation and manual simulation will also be supported by popping up of an execution panel. A set of proposed languages, termed EMASE language (EMASEL) will be developed in accordance with MASE which enhances the modeling power of classical Petri Net. The proposed tool will offer both analytical and simulation solution methods, and is based on classical Petri Nets.

1 Introduction

Petri nets and related graph models have been used in a variety of applications [1]. During the development of net applications, various extensions of Petri nets have been proposed for performance evaluations, measurements and clear, precise representation of the system under consideration. Synchronized Petri Nets (PNs), timed PNs, interpreted PNs, generalize stochastic PNs, continuous PNs, hybrid PNs, color PNs, etc. are examples of the extensions. The main philosophy behind these extensions were to improve their applicability by including exclusive-OR transition [2], switches [2], inhibitor arcs, attributes (e.g. time, capacity, weight, etc.) with components. It is a fact that there is no general formalism of Petri Nets which can be used to fulfill the modeling of various applications and as a result, there will be ongoing introduction of new Petri Net components which may confuse the user eventually.

Due to the powerful formalism and analytical capabilities of Petri nets, a variety of Petri net tools have been proposed [3]. Each of the tool has been developed in a particular environment using different programming languages or even their own languages and editing facilities. The editing of modeled system can be implemented by graphical editor, matrix editor and textual editor. These tools offer different features but one of the many aspects of "user friendliness" seems to be very common among the existing tools. Some of the tools apply extensive mathematical theory and algebra and expect the user to have sufficient technical knowledge while some lack appropriate mapping and graphical user interface capabilities. Further more, the flow of token for some tools are so fast that the user can hardly capture the visualization of each evolution.

After examining the collection of projects related to Petri Nets in [3]. We concluded that a more generic and user-friendly tool is needed. Hence, the create of MASE was undertaken. It is hoped that his package can easily be used by the user through the menu- and mouse-driven interface without reading instruction manuals of the package in details. The implementation of MASE language can help to minimize the introduction of new Petri Net elements.

2 Hierarchical Object-oriented Petri Net (HOOPN)

The underlying Petri Net of MASE is HOOPN. By hierarchical object-orientation, we mean that structurally, there are sub and super layers in the system, which deal with the decomposition of larger system into smaller systems. Each layer is considered as a complete object at its super layer. In MASE, the decomposition is done by applying [Socket Transition]-[Port Place](ST-PP) Interface Substitution Place.

One of the earliest discussion on Hierarchical Petri Nets concept [4] described several hierarchical decomposition of Petri Nets including substitution transition, substitution place, place fusion sets and transition fusion sets. Of these four techniques, substitution place clearly shows the decomposition in terms of object-orientation as place represents state of the model, in which the state can be set to true or false to abstract the cumbersome detail of its sub layer. In contrast, substitution transition shows the instances of the event (transition). Therefore, we only discuss substitution place and alternative ways for this substitution.

The first discussion of substitution place implies the instantiation of the specified sub-page (in [4], page is used instead of layer) and their interactions with the environment via socket transitions. A binding of socket transitions to port transitions in the subpage is specified, where similar name is assigned to both transitions. In other words, the plyg-in metaphor is being used by considering the interacting transition in super page as socket transition and those in sub-page as port transitions. Both sub-and super-pages interact among themselves by merging their respective guards defined by the biding, without taking much care on the aannotation of arcs adjoining socket transition and substitution place. However, this substitution metaphor contains errors like the loss of annotation for the affected arcs and violation to the encapsulation rules, since systematic renaming of identifiers in the substituted model can only be made after all instances of the sub-page [5].

An improved technique to correct the errors mentioned above in substitution place has been introduced by applying the concept of synchronous channel communication extension to Colored Petri Nets [6]. The channel interactions are given by the notations !?ch and ?!ch, which are termed channel output and channel input respectively. With the same biding concept in [4], substitution place is done by seeking the similar name in socket and port transition. Then, modification is done on each transition name to relevant input or output channel with regards to the annotation of two sets of adjoining arcs: those connecting socket transition and substitution place, and those connecting substitution place and port transition. Detailed semantics and examples can be found in [5,6].
The sub and super layers of the substitution places mentioned above interact with each other via the socket transition and port transition respectively. In this case, if sub layers are investigated structurally, port transitions are found to be source transitions in nature, which have no input places and posses output places. However, sub layers are the instances of places at their super layers and can only be instantiated when states of respective super-layer-places become true. With the existence of source transitions in sublayers, there are possibilities that sublayers start their activities autonomously without being instantiated by their super-layer-places.

3 Definition of HOOPN

As mentioned earlier, the underlying Petri Net used by MASE is HOOPN. In addition, interpretation associated with the structure are supplemented to enhance the modeling power. In general, the definition of HOOPN used by MASE is,

HOOPN = (P,T,I,O,M,W,S,E,Text)

Where

P	: a finite set of places
T	: a finite set of transitions
I	: a finite set of input arcs connecting places or input gates to transitions, (PxT)U(SxT)
O	: a finite set of output arcs connecting transitions to places or output gates, (TxP)U(TxE)
M	: marking of the system
W	: weight of the arc
S	: set of input gates of that system
E	: set of output gates of the syste,

Text : interpretation of the system in text form and it contains three elements

Text = (C, Tempo, Op)
Where
C : set of conditions (predicates) associated with transitions
Tempo : timing associated with transitions
Op : set of operations associated with places

4 Graphical User Interface

EMASE provides the following components: graphics editor, drawing validator, text editor, message window, execution panel, subnet creating function, interface function (between text editor and objects on drawing canvas), simulation and analysis function and code generation.

4.1 Graphics Editor

Graphics Editor is a canvas where systems modeled in Petri Nets are to be created. Object selection menu is hidden and brought up only by pressing right mouse button in order to maximize the space of drawing canvas.

4.2 Drawing Validator

This component is built for diagnosis of the created Petri Net model to confirm the correctness of the modeling.

4.3 Text Editor

The interpretations of the Petri Net are written in MASE Language (MASEL) in the text area. MASEL is a formal language developed in accordance with this package. The syntax of the language is parsed by using Lookahead Left-to-right Right-derivation (LALR) parsing algorithm[7, 8].

4.4 Message Window

This window prepares an area for communication between MASE and the users when there is a violation of the formal rules of net theory, occurrence of deadlock or when there is an error occurs.

4.5 Subnet creating function

MASE supports multipaging features. In other words, it allows the user to create a sub-net based on the drawn system in order to enlighten graphical view rather than to throw all the objects in one canvas. This feature can be accomplished by pressing simultaneously the shift key and the first mouse button on the place object which contains sub-net.

4.6 Interface function

Besides all the features mentioned above, we also include the cut, copy and paste features which are important and common in every graphical user interface tool. The user can highlight the interpretation by dragging first button of the mouse on the text area, move to another text area and press the second mouse button. In this way, the highlighted area will be copied to the target window. To cut the highlighted area, the user only needs to press delete key and the whole highlighted area will disappear.

4.7 Simulation and analysis function

Since MASE aims towards producing general and generic Petri Nets, it is important to build certain more technical aspects of the Petri Net theory into the computer system. This makes it possible for the user to apply certain kinds of analysis systems without having a detailed knowledge of the underlying theory. The user decides what is to be done, then the computer performs the involved calculation, which are often very complicated and the computer also checks the validity of the steps proposed by the user.

4.8 Code generator

A code generator is proposed to translate the Petri Net drawings into a pseudo-code. This is to be done after the user has satisfied with the system, interpretations given and simulation observed.

5 Conclusion

This paper discusses MASE, which is a user-friendly software package for the hierarchical object-oriented Petri Net modeling, editing, and analysis of any system at various levels of abstraction. The various hierarchical components of MASE were also discussed. It also emphasized on the expressive power, that can help in drawing a large Petri Nets in a well-structured manner. It supports generalized Petri Nets which can then be extended to model after Color Petri Nets. Besides this extensions, future work on the package will be towards its application in other categories of Petri Net modeling and include their features, firing rules and more net properties, etc.

References

[1] T.Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, vol. 77, No 4, April 1989.

[2] J.L.Peterson, Petri Net Theory and Modelling of Systems, *Prentice-Hall Inc., Englewood Cliffs, N.J. 07632 1981*

[3] Petri Nets Tools, *posted by Groupe ECORP, Centre de Recherche Informatique de Montreal (CRIM), in the Petri nets mailing list*

[4] P. Huber, K. Jensen, R.M.Shapiro, "Hierarchies in Colored Petri Nets", G.Rozenberg(ed.): Advanced in Petri Nets 1990, Lecture Notes of Computer Science, Springer Verlag.

[5] C.A. Lakos, "The Role of Substitution Places in Hierarchical Colored Petri Nets", Technical Reprot TR93-7, Computer Science Department, University of Tasmania (1993).

[6] S.Christensen and N.D.Hansen, "Coloured Petri Nets Extended with Channels for Synchronous Communication", Technical Report DAIMI PB-390. Computer Science Department, Aarhus University (1992).

[7] "lex & yacc, Second Edition", John Levine et. al. O'Reilly & Associates, October 1992.

[8] http://www.devincook.com/goldparser/articles/lalr.htm