Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 15 (2011) 3428 – 3432
Procedia Engineering
www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

A DSPN modeling and simlation Tool for Avionics System

Huobin Tan[a,b] a*, Jiajun Xu[b], Shuzhen Yao[a]

[a]*School of Computer Science and Engineering, Beihang University, 37# XueYuan Road, Beijing 100191, China*
[b]*College of Software, Beihang University, 37# XueYuan Road, Beijing 100191, China*

Abstract

Avionics System is the typical real-time embedded system and DSPN-base system evaluation is the important theoretical basis and supporting technology of the development of it. This paper focuses on the design of the tool based on DSPN, called Aero_DSPN, with the functions of component modelling and simulation analysis of the avionics system, to offer invaluable process and organizational design support. We illustrate system modelling and simulation analysis in a case study of avionics system.

Keywords: avionics system; DSPN; modeling; simulation

1. Introduction

The analytical approach to the evaluation of systems is being increasingly viewed as an integral part of the process of design, analysis, and tuning of computer related field, especially in the avionics systems. Petri nets are commonly viewed as a valid tool for the qualitative and quantitative study of systems. Modeling and analytical models give results whose accuracy depends on the supporting tool, on the designer's ability, and on the level of detail of the model employed. Once the Petri net model has been developed, its solution is generally very quick, so an accurate analysis of the system can be made with the variation of all the model parameters, based on the tool. Petri net models have been employed to solve

* Corresponding author. Tel.: +86-10-82317048; fax: +86-10-82338510.
E-mail address: thbin@buaa.edu.cn.

doi:10.1016/j.proeng.2011.08.642

problems of state observability, system monitoring, and system analysis [1].The focus of the present paper is on the Petri net tool to create and analyze DSPNs (Deterministic and stochastic Petri nets).

The rest of the paper is organized as follows: in Section 2 we introduce the concepts of the DSPN and related work. Section 3 outlines the design and implementation approach. Section 4 provides an avionics systems example which is solved through Aero_DSPN. Conclusions are given in Section 5.

2. Related Work

Petri Nets were introduced by Carl Adam Petri in the early 1960s. Timed Petri Nets (TPNs) are the expansion of Petri nets, by introducing the time delay. Generalized stochastic Petri nets (GSPNs) extends TPN by specifying immediate transitions and exponentially timed transitions. Deterministic and stochastic Petri nets (DSPNs)[7] extend GSPNs by allowing deterministically timed transitions.

A DSPN Petri net is a tuple PN = (P, T, D, G, A, H, I, O, M) where: P is the set of places; T is the set of transitions; D is the determinate variables associate to the transitions; G is the set of random variables associate to transitions; A, and H are the normal and inhibitor Arc; M is the set of marking M: a marking is a tuple, whose cardinality is |P|, recording the number of token in each of the place in P.

In DSPN, the transitions are divided into three types: immediate, discrete, and determinate. The firing time of the immediate is 0; the firing time of the discrete time is generated from the poison random function; the determinate firing time is associated with the period delay.

Many Petri nets modeling tools have been proposed or developed, such as Great SPN 2.0[2], DSPNexpress[3], TimeNet[4], PIPE2.5[5], and so on. All these tools have the following characteristics in common: free of charge for academia, Graphical User Interface (GUI) and modeling of GSPN. Other important features that are not shared between these tools are summarized in Table 1.

Table 1. Petri Nets tool Comparison

Features	GreatSPN 2.0	DSPNexpress	TimeNet 4.0	PIPE 2.5
DSPN supported	No	Yes	Yes	No
PNML based	No	No	No	Yes
Component Modeling	No	No	No	No
Simulation	Yes	No	Yes	No

Unfortunately, many of them have problems including: serious functionality problems that hamper usability, lack of support of DSPN and for PNML[6], and a lack of analysis power necessary to conduct both qualitative and quantitative analysis on DSPN. Rarely, they have the support of component modeling.

In this paper we present a new Petri net modeling tool for the analysis and simulation of the avionics system. This tool, called Aero_DSPN is extended from PIPE2.5 and has been extended with the functions of the component modeling, simulation and numerical analysis of DSPN. It provides a discrete and determinate time approximation which results in the possibility to analyze a DSPN mode.

3. The Design and Implementation Approach

3.1. System Architecture

In this section, we give an overview of Aero_DSPN. The overall architecture is shown in Fig.1(a). The two main layers are the Graphical User Interface (GUI) and the PNMLData. The package diagram is shown in Fig.1(b). The GUI is built on top of the general graphical package Swing, and DataLayer and Components are built on top of the PNMLData, When a DSPN model has been created in the GUI Editor, the Syntax Checker of the compiler is invoked to ensure that the model constitutes a legal DSPN model. The Component Modeling provides a function of modeling sub-DSPN. Enabling and occurrence of transitions are calculated by the simulation of the DataLayer, and the simulation is controlled and viewed by the user in the Simulation of the GUI.

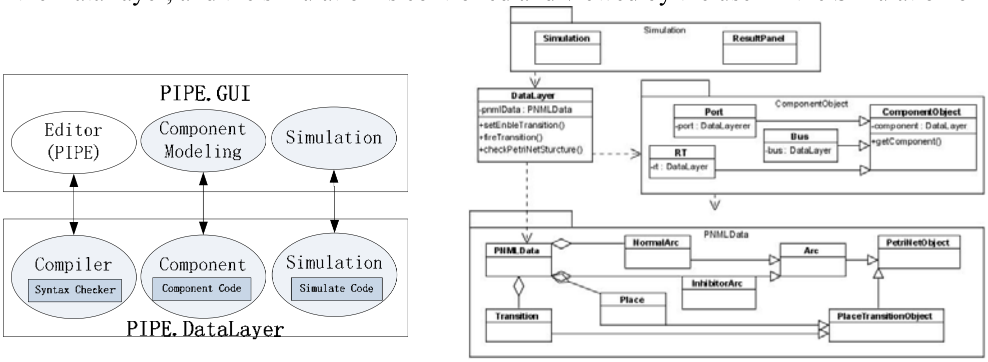

Fig. 1. (a) DSPN architecture; (b) package diagram

- DataLayer(PNMLData)

The PNMLData Package contains all the classes used to represent a Petri net. The Petri net model is made up of Places, Arcs, and Transitions. Places, Arcs and Transitions each have several attributes that characterize their properties, such as id, name, location, etc. These attributes are stored in variables internal to each object. The elements are inheritance of the PetriNetObject. Each Petri net is encapsulated by an instance of the PNMLData class. The PNMLData class contains all the Petri net Objects stored in ArrayLists. PNMLData has not only methods to access all its internal objects and to return its internal lists, but also methods to calculate the current marking, initial marking, and enabled transitions.

- GUI

All the parts of the GUI extend Java Swing components, enabling them to take advantage of the base class features, to create the visual representation of the Petri net elements (place, transitions, and arcs).

There is a syntax check function in GUI, to ensure the legal constitution of the DSPN model. The GUI editor enforces a number of built-in syntax restrictions, and thereby prevents the user from making certain syntax errors during the construction of a model.

The PNMLData data and the GUI are both updated frequently and must remain synchronized, to make the PNMLData data and visual representation consistent. PNMLData is used as the model, storing the data, whilst the GUI viewer creates the visual representation of the component from the data in the model and handles user interactions.

3.2. Components Modeling

Some Petri net modeling tools have been used to model and simulate the models, but a domain-oriented (component-oriented) modeling and simulation tool have not been produced yet for avionics systems. We introduce domain-oriented (component-oriented) modeling function.

Component modeling offers a proven approach to drive a specialized focus on the avionics system. A DSPN component is a DSPN equipped with input places and output places, which represent the interface of the component to other components. Each component is a composition of the input places, output places, transitions, and Arcs as show in Fig.2(a). By adding explanations to component operations, we would have a precise representation of operations for each component. We create the sub-models of the

1553B avionics system, as show in Fig.2(b). The components are constructed from instances of these components. The whole DSPN model is merged from these component models. The 1553B avionics bus system is divided into three components, which are described in the following.

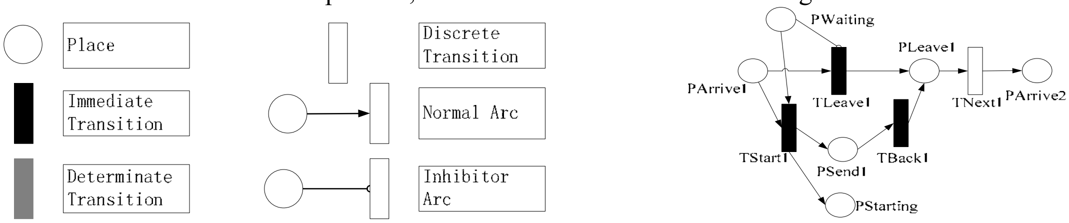

Fig. 2. (a) DSPN elements; (b) port component

- RT component: Remote terminals are the largest group of bus components. RT component stands for the periodic creation of the message, as Fig.3a). The frequency of periodic data transmissions is ensured by a timer-based preemption mechanism at every terminal. The aperiodic message is configured by setting the transition properties. The configuration of a terminal with periodic data or a periodic is operated by setting the properties of transition T_{Begin}.
- Bus transport component: In 1553B bus, as Fig.3b), each order contains multiple data, whose max size 32. The delay time of the transport is depended on the number of the data.
- Port component: The port component as Fig.2(b) is a center is a waiting center, where the message has been created and is located here waiting to be transport onto the bus. The 1553b bus uses the mechanism of Polling to get the message from the Port.
- Components Merge: To extend the benefit of component technology into the whole design lifecycle of the DSPN model, proposals for component merge in the creating of the model have started to emerge. We define the place position merge operation that leads to the main result of this section. In view of place merge, we define an operation called merge that combines the places of two different subnets.

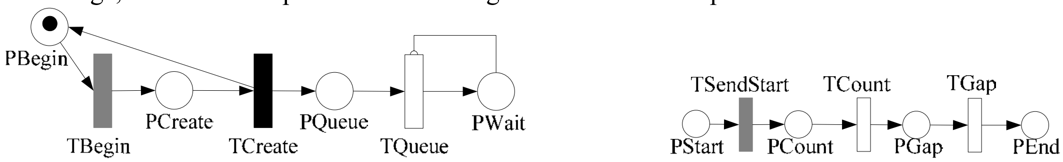

Fig. 3. (a) rt component; (b) bus transport component

4. Avionics System Example

In order to verify 1553B avionics system, with high-speed, low latency and advanced performance, we will do modeling based on our tool as shown in Fig.4(a).

The core module of the avionics system is Data Process and Display System (DPDS) in the example. Meanwhile, it contains four external cross-linked subsystems: the integrated navigation system (INS); weapons fire control and store management system (ASMPS); Air Data System (ADS); timekeeping device (ST). In this system, DPDS is treated as a bus controller (BC); INS, ADS, and ST are treated as remote terminal (RT). Data between RT and BC is transmitted by using the 1553B transmission protocol. Table 2 shows the data exchange between various systems and transfer cases.

Here, we would like to take a 1553B avionics system with 7 RTs. By using the DSPN 1553B components, we create the DSPN model in the following steps: firstly, create the RT components, the port components for each RTs, and an Bus transport component; then, Perform the Place merge to composite the DSPN model from each sub components; then, modify the model, if necessary; finally, update the token in the DSPN model. By using the above steps, The system DSPN is created.

Because the transfer rate of Fiber Channel is 1Gbps, a full load of the data frame transmission time is 0.002 148ms. In the Areo_DSPN, we set single step 1ms, and speed 1. We perform the simulation and get bus utilization and the delay time, as shown in Fig.4(b).

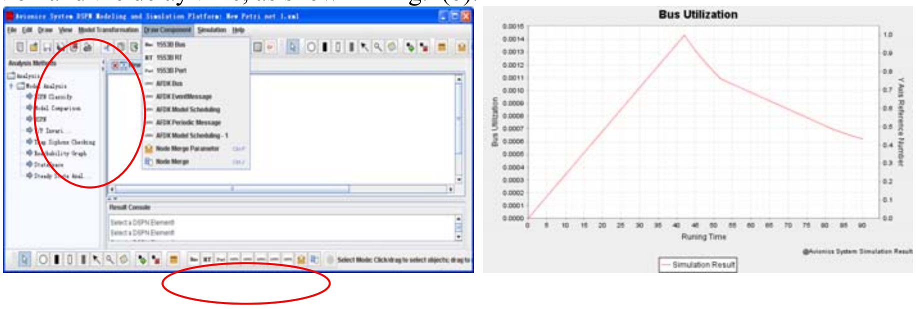

Fig. 4. (a) Areo_DSPN features; (b) bus utilization

Table 2. System Data

Sources	Data	Cycle	Delay	Protocol	Destination
INS	ALL	T/40ms	20ms	1553B	DPDS
ADS	ALL	T/40ms	20ms	1553B	DPDS
ST	DATE/TIME	T/500ms	20ms	1553B	DPDS
ASMPS	INFOMATION	T/40ms	20ms	1553B	DPDS

5. Conclusion and Future work

The future work would focus on the transformation from UML to DSPN. Aero_DSPN could help to do the performance evaluation of the system and business UML model could easily be created by the user. We need to transform the avionics system business UML model to DSPN analysis model.

Acknowledgements

The research is supported by Science and Technology on Avionics System Integration Laboratory under grant No. 20085551042.

References

[1] Sahika Genc, Stphane Lafortune. IEEE Transactions Automation Science and Engineering. vol. 4, no. 2, pp. 206-219, 2007.

[2] GreatSPN 2.0. GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets. http://www.di.unito.it/~greatspn/.

[3] DSPNexpress, Open Source Toolkit for Performance Evaluation of UML System Specifications. http://www.dspnexpress.de/.

[4] TimeNET 4.0. TIMEd Net Evaluation Tool. http://pdv.cs.tu-berlin.de/~timenet/.

[5] P. Bonet, CM Llado, R. Puijaner, and WJ Knottenbelt. PIPE v2.5. http://pipe2.sourceforge.net/.

[6] Petri Net Markup Language. http://www2.informatik.hu-berlin.de/top/pnml/about.html.

[7] Molly M.Discrete time stochastic Petri nets[J].IEEE Trans on Software Engineering,1985,31(4):417-423.