Integration of analytic model and simulation model for analysis on system survivability

Jang Se Lee

Department of Computer Engineering,
Korea Maritime and Ocean University, Busan, Korea

Summary
The objective of this paper is to present a method for integrating Stochastic Petri Net model as an analytic model into Discrete Event System Specification model as a simulation model. To do this, we have made place and transition which are components of Stochastic Petri Net to Discrete Event System Specification based models respectively. And we have produced a structure for simulation by coupling between place models and transition models according to a structure of Stochastic Petri Net. The proposed method is validated by comparison with results obtained from analytic analysis of Stochastic Petri Net model and from simulation of transformed Discrete Event System Specification model.

Key words:
Integration, Analytic model, Simulation Model, SPN, DEVS

1. Introduction

A Stochastic Petri Net which is a representative method using analytic model has been used in various areas for analysis of performance, availability, reliability, and survivability, etc. However it is difficult to extend to analyze large scale system because of the state space explosion problem [1]. To overcome this problem, simulation approach would be able to be applied as one of alternatives [2,3] but so far there have not existed researches using theoretically well-defined formalism.

In this paper, we present a method for integrating Stochastic Petri Net model into Discrete Event System Specification model as a representative formalism for simulation. To do this, we have made a Discrete Event System Specification based models of place and transition respectively which are components of Stochastic Petri Net formalism and have produced Discrete Event System Specification based model structure for simulation by coupling between place models and transition models according to structure of Stochastic Petri Net.

We validate our proposed method by comparison with results obtained from SPNP package [3] which is a tool for analytic analysis of Stochastic Petri Net model and from simulation of Discrete Event System Specification model transformed by the proposed method.

Manuscript received July 5, 2015
Manuscript revised July 20, 2015

This paper is organized as follows. In Chapter 2, we review Stochastic Petri Net and Discrete Event System Specification as a background. Chapter 3 proposes the integration from Stochastic Petri Net to Discrete Event System Specification, In Chapter 4, we validate our proposed method by comparison with analysis results of each model. Finally, we conclude this work in Chapter 5.

2. Background

2.1 Stochastic petri net

A Stochastic Petri Net (SPN) which is extension of a Petri net is modelling formalism for the automated generation of Markovian stochastic systems. The details of SPN may be found in [4,5].

A SPN has 5-tuples;

A = (*P*, *T*, A, R, *M'*)

Where

P : set of places
T : set of transitions
A : input arcs and output arcs
R : set of firing rates (with marking-dependent) associated with transitions
M' : initial marking

2.2 Discrete event system specification formalism

A Discrete Event System Specification (DEVS) is the formalism for simulation. The details of DEVS may be found in [6,7]. A DEVS has models of two types which are atomic model and coupled model.

An atomic model has 7-tuples;

AM = (X, S, Y, δ_{int}, δ_{ext},λ, ta)

Where

X : set of external events(input)
S : set of sequential states
Y : set of outputs
δ_{int} *: internal transition function*
δ_{ext} *: external transition function*
λ *: output function*
ta: time advance function

A coupled model has 5-tuples;

CM = (D, $\{M_i\}$, $\{I_i\}$, $\{Z_{i,j}\}$, *select*)

Where

D: set of component names

M_i *: basic component model*

I_i *: influence set, for each j in Ii*

$Z_{i,j}$ *: coupling set of i-to-j output translation*

select: tie-breaking selector function

3. Integration from SPN to DEVS

To integrated SPN into DEVS, firstly, we convert place and transition of SPN into each DEVS based atomic model according to the behavioural nature of them. Secondly, tuples except place and transition in SPN are applied as attributes of a place model and a transition model and information for coupling. Thirdly, we connect place models and transition models together according to input and output relations of SPN. Finally, we obtain a coupled model structure for simulation.

3.1 Place modeling

A place is converted to a DEVS based atomic model. Fig 1 shows state diagram of place model. In Fig 1, circles mean states, dotted arrows mean internal transition and output due to internal transition, solid arrows mean external transition and input as an external event, the triangle of left side means input ports and the triangle of right side means output ports, and contents in the box of upper corner mean representative attributes.

A place plays role of two types. The one is an input place connected to transition with input arc and the other is output place connected from transition with output arc. When input places of a transition have the required number of tokens, the transition is enabled. And if an enabled transition fires, specified tokens are removed from input places and are deposited to output places.

To model this, a place model has four states which are ‘checking’ as an initial state, ‘empty’, ‘nonempty’, and ‘output’. When a state of a place model is ‘checking’, a place model checks whether it has the required number of tokens or not. If a place model has the required number of tokens, it sends ‘enable (marking)’ message to the transition model connected to the input place model and changes to ‘nonempty’ state. If not, it sends ‘disable’ message to the transition model connected to the input place model and changes a state to ‘empty’. When a state of a place model is ‘nonempty’, if a place model receives a ‘fire’ message, it changes to ‘output’ state and it sends a specified number of tokens or ‘disable’ message to output place model via transition model. When a state of a place model is ‘empty’ or ‘nonempty’, if it receives a token, it changes its state to ‘checking’.

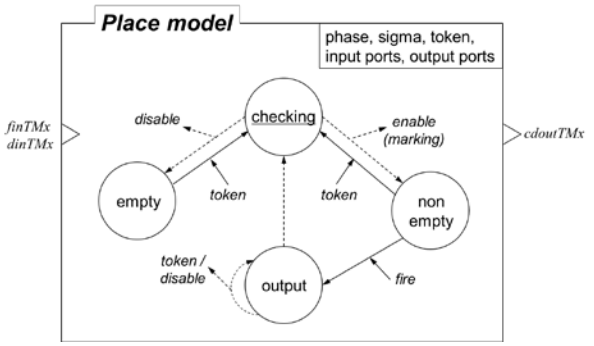

Fig. 1 State diagram of DEVS based atomic model for place

3.2 Transition modeling

A transition is also converted to a DEVS based atomic model. Fig 2 shows state diagram for transition. In Fig 2, each figure’s meaning is same with them in Fig 1.

A transition is enabled when input places connected with the transition have the required number of tokens, and an enabled transition may fire after sojourn time which is decided by exponentially distributed rate elapses. In the case of marking dependent rate, sojourn time is decided by marking of the input places associated with a transition.

To model this, a transition model has five states which are 'checking' as an initial state, ‘disabled’, ‘enabled’, ‘input’, and ‘output’. When a state of a transition model is ‘checking’, a transition model checks whether it is enabled or not. If a transition model receives ‘enable (marking)’ message from all of its input place models, it changes to ‘enabled’ state. If not, it changes to ‘disabled’ state. A sojourn time at ‘enabled’ state is decided by exponentially distributed rate associated with transition model and marking received from its input place model. When a state of a transition model is ‘disabled’, if a transition model receives a ‘enable (marking)’ or ‘disable’ message, it changes to ‘checking’ state. And if it receives a ‘enable (marking)’ or ‘disable’ message at ‘enabled’ state before firing, it adds the time being passed from the time being changed to ‘enabled’ state to the elapsing time of transition model and changes its state to ‘checking’ state.

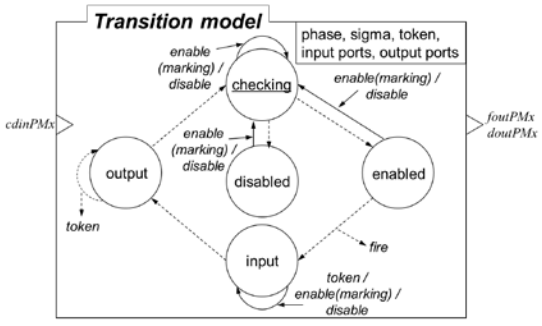

Fig. 2 State diagram of DEVS based atomic model for transition.

After sojourn time of transition model elapses, a transition model sends 'fire' message to each of input place models connected with it and changes its state to 'input'. If a transition model receives tokens from each of its input place models at 'input' state, it changes to 'output' state and sends tokens to each of its output place models. After sending, it changes 'output' state to 'check' state. In case of receiving 'enable (marking)' or 'disable' message at 'input' state or 'output' state, it saves the message to decide whether it is enabled or not.

3.3 Coupling between models

We have connected between place models and transition models with relation of input arcs and output arcs. An input arc is used to connect from an input place model to a transition model and an output arc is used to connect from a transition model to an output place model.

Table 1 shows the information of ports for coupling between a place model and a transition model. For example, a place model (PMx) has three ports, which are cdoutTMx, finTMx, and dinTMx. The cdoutTMx means a name of a port and it is connected from a place model (PMx) itself as an input place to a transition model (TMx). And 'enable (marking)' or 'disable' message or tokens means the contents which are transferred from PMx model to TMx model. As another example, a transition model (TMx) has three ports, which are cdinPMx, foutPMx, and doutPMx. The doutPMx means a name of a port and it is connected from a transition model (TMx) itself to a place model (PMx) as an output place. And tokens mean the contents which are transferred from TMx model to PMx model.

4. Case study

We validate our proposed method by comparison with analysis results between each model through the workstation-file server (WFS) example [8].

As shown in Fig. 3, we consider a system consisting of two workstations and one file server. These devices are connected by fault-free network. The system is operational so long as one of workstations and one of file server are operational. And each device is repairable but file server has priority over workstations. This means that a file server is repaired before repairing a workstation if a workstation and a file server are failed at a same time. We assume exponentially distributed times to failure and repair. And let λ_w, μ_w, λ_f, and μ_f denote failure rate of a workstation, repair rate of a workstation, failure rate of a file server, and repair rate of a file server respectively.

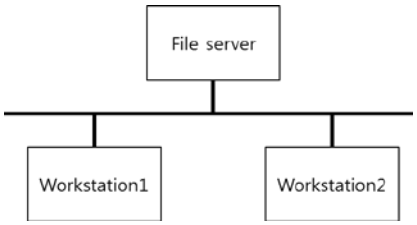

Fig. 3 System architecture of 2-workstation and 1-file server

Fig. 4 shows the availability model using SPN of WFS example shown in Fig. 3. Place P_{wsup} and two tokens (denoted by black dot) represent two of working workstations. Place P_{fsup} and a token represent a working file server, while Place P_{wsdn} and P_{fsdn} represent a failed workstations and a failed file server. Transition T_{wsfl} represents a failure of workstations and its firing rate depends on marking of Place P_{wsup} which means number of working workstations and Transition T_{fsfl} represents a failure of a file server. Transition T_{wsrp} and T_{fsrp} represent a repair of workstations and a repair of a file server. Transition T_{wsrp} can fire when a file server is working because T_{wsrp} is connected to Place P_{fsup} and P_{wsdn}.

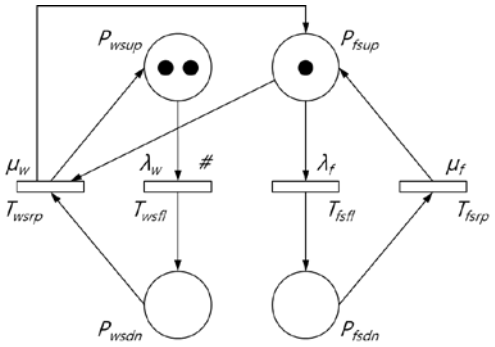

Fig. 4 SPN model for the WFS example shown in Fig.3

Table 1: Ports of each model for coupling

Model	Port	From	To	Contents
Place model (PMx)	cdoutTMx	Place PMx (input place)	Transition TMx	enable(marking) or disable message / tokens
	finTMx	Transition TMx	Place PMx (input place)	fire message
	dinTMx	Transition TMx	Place PMx (output place)	tokens
Transition model (TMx)	cdinPMx	Place PMx (input place)	Transition TMx	enable(marking) or disable message / tokens
	foutPMx	Transition TMx	Place PMx (input place)	fire message
	doutPMx	Transition TMx	Place PMx (output place)	tokens

We transformed the SPN model shown in Fig. 4 into simulation structure by coupling DEVS models proposed in Chapter 3. Fig. 5 shows the simulation structure. As shown in Fig. 5, the simulation structure is composed of four place models and four transition models like as places and transitions in the SPN model shown in Fig. 4 and each model is coupled to each other according to arcs in the SPN model shown in Fig. 4.

To validate the proposed method, we compared the analysis results of steady-state availability on WFS example by analytic and simulation method. Table 2 shows parameters for analysis.

Table 2: Parameters for analysis

Parameters	Value
λ_w	0.0003
μ_w	1.0
λ_f	0.0001
μ_f	1.0

We evaluated the analytic model using SPNP packages [3] and simulated the integrated model using DeSim engine [9]. We replicated this simulation five times for simulated time of 525600 per run and applied a point estimation of 95% confidence intervals. The steady-state availability for each model of the case study was presented in Table 3.

Through comparison of results in Table 3, we can know that the simulation result from the integrated DEVS model has nice match with the analytic result from the SPN model and that proposed approach is suitable to integrate analytic model into simulation model.

Table 3: Comparison of steady-state availability between each method

Analysis method	Availability
Analytic method (SPN model)	9.998998E-01
Simulation method (DEVS model)	9.999002e-01 [9.998789e-001, 9.999215e-01]

5. Conclusions

In this paper, we have proposed a method for integrating SPN model as an analytic model into DEVS model as simulation model. To do this, we have performed modeling of tuples being in SPN formalism according to behavioral nature of them. A place model and a transition model were presented as the result of modeling. SPN models are transformed to the coupled model structure for simulation by connecting proposed place models and transition models together according to arcs in SPN models. We have shown that proposed approach is suitable through comparison of the analytic and simulation result on case study of WFS example.

As a future work we will extend proposed method with characteristics of SRN for allowing more convenient modeling.

Acknowledgments

This work was partially supported by the Research fund of Survivability Technology Defense Research Center of Agency for Defense Development of Korea (No. UD120019OD) and the fund of the Korea Maritime and Ocean University.

The author would like to thank Prof., Dr. Kishor. S. Trivedi for his valuable advice.

References

[1] V. Mainkar and K.S. Trivedi, "Fixed point iteration using stochastic reward nets", Proceedings of the Sixth International Workshop on. IEEE, pp. 21-30, 1995.

[2] G. Ciardo and K.S. Trivedi, "A decomposition approach for stochastic reward net models", Performance Evaluation, Vol.18, pp. 37-59, 1993.

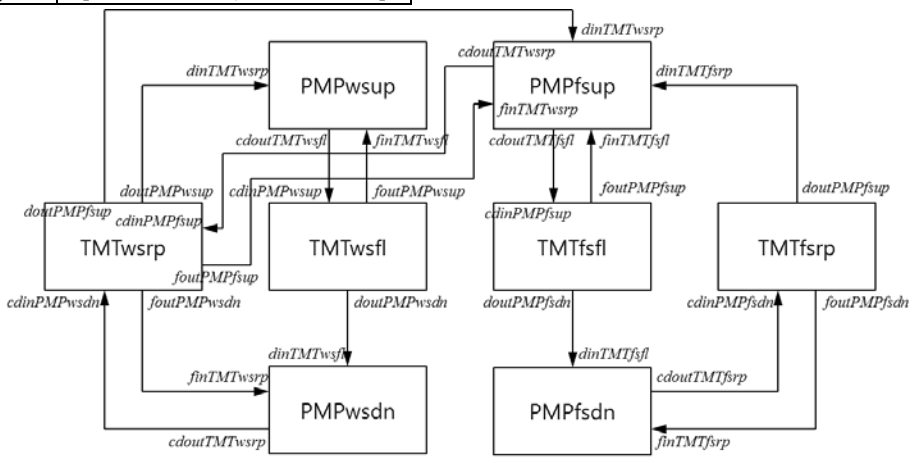

Fig. 5 Integrated simulation structure for SPN model shown in Fig.4

[3] C. Hirel, B. Tuffin, and K.S. Trivedi, “SPNP: Stochastic petri nets. Version 6.0.”, Computer Performance Evaluation, Modeling Techniques and Tools, Springer Berlin Heidelberg, pp. 354-357, 2000.
[4] M.A. Marsan, G. Balbo and G. Conte, “A Class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems”, ACM Transactions on Computer Systems, Vol.2, No.2, pp. 93–122, 1984.
[5] M.K. Molloy, “Performance analysis using stochastic petri nets”, IEEE Trans. on Comp., Vol.31, pp. 913-917, 1982.
[6] B.P. Zeigler, Object-oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic systems, Academic Press, 1990.
[7] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic press, 2000.
[8] K.S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Application, 2nd ed., Wiley, 2002.
[9] Y.K. Kim, Model-based Design for Intelligent Systems: Intelligent Card Game Player, Master degree thesis, Korea Aerospace University, 1997.

Jang Se Lee received the B.S., M.S., and Ph.D. degrees in Computer Engineering from Korea Aerospace University in 1997, 1999, and 2003 respectively. During Jan. 2013 - Jan. 2014, He stayed in Duke High Availability Assurance Lab (DHAAL), Duke University as a visiting scholar. He is currently an associate professor (from 2004) in the division of Information Technology, Korea Maritime and Ocean University. His research interest includes modeling and simulation, intelligent systems, security for systems and networks, system survivability, and E-navigation.