

## Research Article

# A Key Event Path Analysis Approach for Integrated Systems

**Jingjing Liao**

*China Ship Development and Design Center, 268 Zhang Zhidong Road, Hubei Province, Wuhan 430064, China*

Correspondence should be addressed to Jingjing Liao, [jjleap@gmail.com](mailto:jjleap@gmail.com)

Received 6 April 2012; Accepted 23 July 2012

Academic Editor: Hing Kai Chan

Copyright © 2012 Jingjing Liao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By studying the key event paths of probabilistic event structure graphs (PESGs), a key event path analysis approach for integrated system models is proposed. According to translation rules concluded from integrated system architecture descriptions, the corresponding PESGs are constructed from the colored Petri Net (CPN) models. Then the definitions of cycle event paths, sequence event paths, and key event paths are given. Whereafter based on the statistic results after the simulation of CPN models, key event paths are found out by the sensitive analysis approach. This approach focuses on the logic structures of CPN models, which is reliable and could be the basis of structured analysis for discrete event systems. An example of radar model is given to characterize the application of this approach, and the results are worthy of trust.

## 1. Introduction

The American Department of Defense published the “Department of Defense Architecture Framework (DoDAF) Vision 2.0” in 2009 [1–3], which concludes 41 models and 8 viewpoints: All Viewpoint (AV), the Capability Viewpoint (CV), the Data and Information Viewpoint (DIV), the Operational Viewpoint (OV), the Project Viewpoint (PV), the Services Viewpoint (SvcV), the Standards Viewpoint (StdV), and the Systems Viewpoint (SV) and develops one of the main standards in the world on account of its integrated description criterion for system architecture. However, the static description could not support the dynamic performance evaluation of the integrated system, which leads lots of researchers to study the problem of creating executable models from the DoDAF products [4–6] (Figure 1). As a kind of high level Petri net, CPN is the preferred tool to create executable models for its beautiful advantage in modeling, simulation, and performance evaluation of large scale systems.

CPN adopts color sets, class folding, and definitions of data class, which was founded by Professor Jensen [7] in 1981 and enforces the modeling ability of Ordinary Petri Net (OPN). There have developed many mature theories about modeling manners and simulation results based on CPN [8–10], but the structural characteristics such as incidence

matrices are weakened from OPN to CPN, which prevents us to obtain the S (T) invariants, trap (siphon) structures, conservation, and so forth based on incidence matrices. As a result, we have to search for the structural analysis methods of CPN models.

The structural analysis [11] is a kind of efficient performance analysis method, which has the advantage of obtaining the logic verification results from the whole structures of the models independent of their initial markings and getting the behavior characteristics from the structures only. Through the structural analysis of CPN models, we mainly want to indicate the events and choice relations between them from their structures. What is more, we should find out the key event paths if needed in project.

Events and choice relations between them are on the basis of event structures. A probabilistic event structure [12] is the probability extension based on the extended bundle event structure, which we can use to represent the concurrent operation and interrupt operation in Process Algebra. In this paper, we adopt the probabilistic event structure graphs translated from the architecture models and CPN models to analyze the serial event paths and the cycle event paths. According to the probability range for each cycle event path based on the simulation results of the associated CPN model, we can determine the event path with the maximum change range to be the key event path of the system.

TABLE 1: DoDAF V2.0 OV models.

Models	Descriptions
OV-1: high-level operational concepts graphic	The high-level graphical/textual description of the operational concept.
OV-2: operational resource flow description	A description of the resource flows exchanged between operational activities.
OV-3: operational resource flow matrix	A description of the resources exchanged and the relevant attributes of the exchanges.
OV-4: organizational relationships chart	The organizational context, role, or other relationships among organizations.
OV-5a: operational activity decomposition tree	The capabilities and activities (operational activities) organized in a hierarchal structure.
OV-5b: operational activity model	The context of capabilities and activities (operational activities) and their relationships among activities, inputs and outputs; additional data can show cost, performers, or other pertinent information.
OV-6a: operational rules model	One of three models used to describe activity (operational activity). It identifies business rules that constrain operations.
OV-6b: state transition description	One of three models used to describe operational activity (activity). It identifies business process (activity) responses to events (usually, very short activities).
OV-6c: event-trace description	One of three models used to describe activity (operational activity). It traces actions in a scenario or sequence of events.

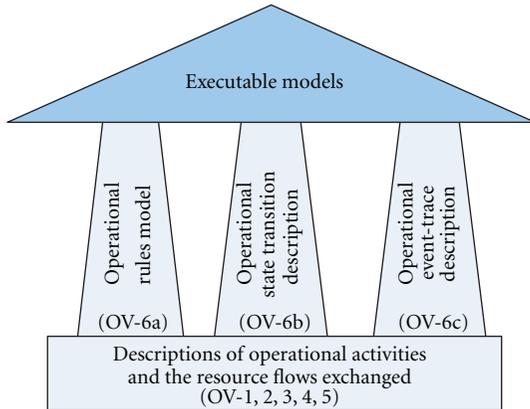


FIGURE 1: Executable models based on DoDAF-OV models.

## 2. Basic Concepts

*2.1. OV of DoDAF 2.0.* Like the previous DoDAF visions, DoDAF V2.0-described Models in the Operational Viewpoint describe the tasks and activities, operational elements, and resource flow exchanges required to conduct operations. They definitely define operational activities and their relationships and provide efficient references [2]. Names of the models and their descriptions are provided in Table 1.

The OV-1 describes a mission, class of mission, or scenario. The OV-2 applies the context of the operational capability to a community of anticipated users. The OV-3 addresses operational resource flows exchanged between operational activities and location. The OV-4 shows organizational structures and interactions. The OV-5 describes the operational activities that are being conducted within

the mission or scenario. The previous OV models merely describe the static structure of the architecture elements and their relationships. However, OV-6 focuses on the dynamic behavior, which concerns the timing and sequencing of events that capture operational behavior of a business process or mission thread. Thus, this behavior is related to the activities of OV-5b. Based on those static and dynamic description models, we can create the associated simulation models (e.g., CPN models).

*2.2. Basic Definitions of Event Structures.* Event structures are a prominent noninterleaving model for concurrency. We use Langerak’s bundle event structures [13], an adaptation of Winkel’s event structures [14] to fit the specific requirements of multiparty of synchronization. Bundle event structures consist of events labeled with actions—an event modeling the occurrence of its action—together with relations of causality and conflict between events. System runs can be modeled as partial orders of events satisfying certain constraints posed by the causality and conflict relations between the events.

Causality is represented by a relation, denoted  $\mapsto$ , between a set  $X$  of events and an event  $e$ . The interpretation of  $X \mapsto e$  is that if  $e$  happens in a system run, exactly one event in  $X$  has happened before (and cause  $e$ ). This enables us to uniquely define a causal ordering between the events in a system run. Conflict is a symmetric binary relation, denoted  $\#$ , between events. The intended meaning of  $e\#e'$  is that either  $e$  or  $e'$  that either  $e$  or  $e'$  can appear in a system run but not both.

*Definition 1.* A bundle event structure (BES)  $\varepsilon$  is defined to be a structure  $(E, \#, \mapsto, l)$  with  $E$  a set of events,  $\# \in E \times E$  the

(irreflexive and symmetric) conflict relation,  $\mapsto \subseteq P(E) \times E$  the causality relation, and  $l : E \rightarrow L$  the action-labelling function, where  $L$  is a set of action labels, such that

$$\forall X \subseteq E,$$

$$e \in E : X \mapsto e \implies (\forall e', e'' \in X : e' \neq e'' \implies e' \# e''). \quad (1)$$

We usually denote bundle  $(X, e)$  by  $X \mapsto e$  and an event labeled  $a$  by  $e_a$ . But it is difficult to simulate a system's behavior only depending on the sequences of activities. For concurrent systems, the performance and security analysis is a more important problem. So it is necessary to consider some performance evaluation parameters (e.g., probability metrics). In the following part of this paper, we will obtain probabilistic extended bundle event structures ( $\pi$ EBESs) by decorating events and bundles with probabilities.

In  $\pi$ EBESs, we bind probabilities to some events, which we denote probabilistic events. And we associate a probability distribution time value with some events, which are random timed events. Then we can obtain generalized stochastic extended bundle event structures. We use a partial order map  $\pi$  to represent the probability of an event, then  $\pi(e) = p(p \in [0, 1])$  expresses that the probability is  $p$  if the event  $e$  is enabled.

*Definition 2.* A  $\pi$ EBES  $\varepsilon$  is defined to be a structure of  $\varepsilon = (E, \#, \mapsto, l, \pi)$  with  $E$  a set of events,  $\# \in E \times E$  the (irreflexive and symmetric) conflict relation,  $\mapsto \subseteq P(E) \times E$  the causality relation, and  $l : E \rightarrow L$  the action-labelling function,  $\pi : E \rightarrow_p (0, 1)$  the probability function,  $\rightarrow_p$  the partial order function. For each event  $e$ , there exists  $e \in \text{dom}(\pi)$ ,  $\exists Q \subseteq \text{dom}(\pi) : e \in Q \wedge Q$  such that  $\bigwedge \sum_{e' \in Q} \pi(e') = 1$ .

$\pi$ EBESs are graphically represented as PSEGs in the following way: events are denoted as dots; near the dot the action label is given. Conflicts are indicated by dotted lines between representations of events. Bundle  $X \mapsto e$  is indicated by drawing an arrow from each event in  $X$  to  $e$  and connecting all arrows by small lines. The probabilities and delays are labeled beside the associated events. If there exists a cycle causality, we use a dash-dotted line to instead of the small line for bundle  $X \mapsto e$ . The basic relations are as Figure 2.

### 3. Key Event Path Analysis

At present, there have been many research results and theories about events and route choices. Based on these, we can translate CPN models to PSEGs according to some rules to analyze event paths of systems. In this section, the translation rules and event path definitions are presented. Besides that, the key event path analysis method based on the simulation results of CPN models is brought forward.

*3.1. PSEGs Translation Rules and Steps.* Before discussing the translation rules of PSEGs, we should study the contents represented by DoDAF V2.0-OV models and the relationships between them. As referred in Section 2.1, OV-1

to OV-5 mainly depict the operational activities and resource flows exchanged, and OV-6 adds the dynamical behavior descriptions of the operational activities. In other words, the activity and node descriptions in the models of OV-1 to OV-5 are associated to each other, which are the basis of the event sequences and time delays in OV-6 (as in Figure 3).

An event models the occurrence of its action (very short activity). As in Table 1, OV-5a and OV-5b represent the activities and behavior relationships between them when occurring, the associated events of which makes up the events of PSEGs. And OV-6a provides instructions which activities should occur under some certain conditions. OV-6b denotes the activity sequences, and OV-6c the information exchanged between all operational nodes. All OV-6 models can determine the relationships between events in PSEGs to be causality, conflict, or independence. In conclusion, we can pick up event elements from OV-5 models and relationships from OV-6 models for PSEGs. Then the probabilities and time delays of occurring events are left unknown (the ones provided by OV-6 may be denoted by some random distributions, which is not convenient to compute and analyze), which can be obtained from the statistical simulation results of CPN models.

As a popular tool for modeling, simulating, and analysis, CPN tools support random distribution functions, and times of simulation, which enables us to pick up favorite statistical data to analyze. When simulating a CPN model, it is often useful to be able to examine the markings and occurring binding elements, to periodically extract information from the markings and binding elements, and then to use the information for different purposes. A monitor is a mechanism in CPN tools that is used to observe, inspect, control, or modify a simulation of a CPN model. Many different monitors can be defined for a given net. Monitors can inspect both the markings of places and the occurring binding elements during a simulation, and they can take appropriate actions based on the observations. Monitors can be used for each of the activities. Therefore, we can conclude the PSEGs translation rules as in Figure 4.

On the basis of the  $\pi$ EBESs definition, DoDAF-OV models and CPN models, we can conclude the particular steps as follows.

*Step 1.* Extract the bottom activities to construct event sets  $E = \{e_i\}$ , graphically denoted as dots.

*Step 2.* Make sure the relationships between relationships.

*Substep 1.* The relationship of event  $e_1$  and  $e_2$  as the form of "if  $e_1$  then  $e_2$ " is if  $e_1$  occurs before  $e_2$  in the event sequence description of OV-6b and OV-6c, then the relationship of them is a cycle causality, graphically denoted as a dash-dotted arrow from  $e_1$  to  $e_2$ , or the relationship is a causality, graphically denoted as an arrow from  $e_1$  to  $e_2$ .

*Substep 2.* The relationship of  $e_2$  and  $e_3$  as the form of "if  $e_1$  then  $e_2$ , else  $e_3$ " in OV-6a is conflict, graphically denoted as a dashed line connecting with  $e_2$  and  $e_3$ .

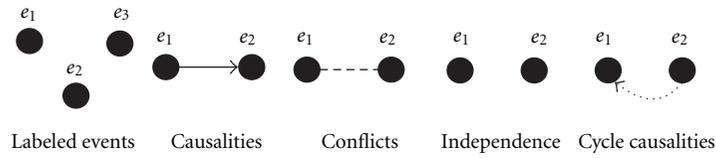


FIGURE 2: Basic relations of  $\pi$ EBESs.

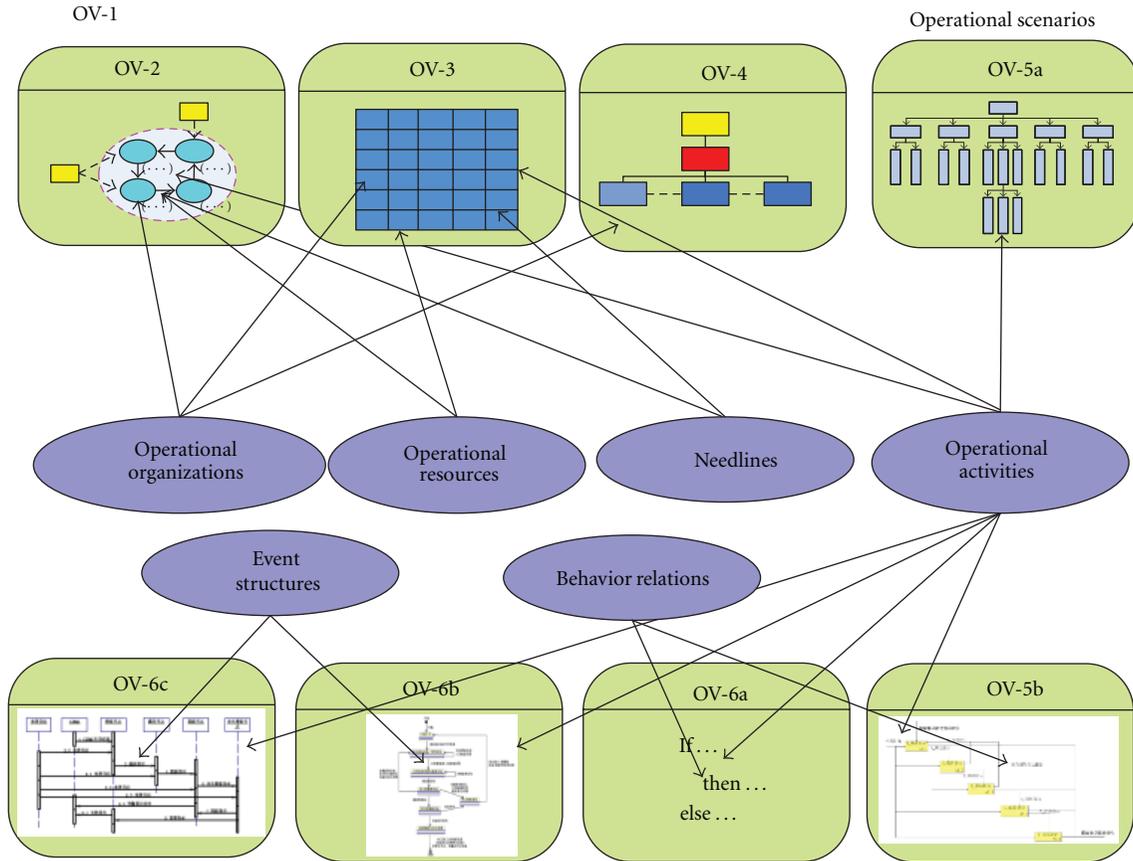


FIGURE 3: Relationships between DoDAF v2.0 models.

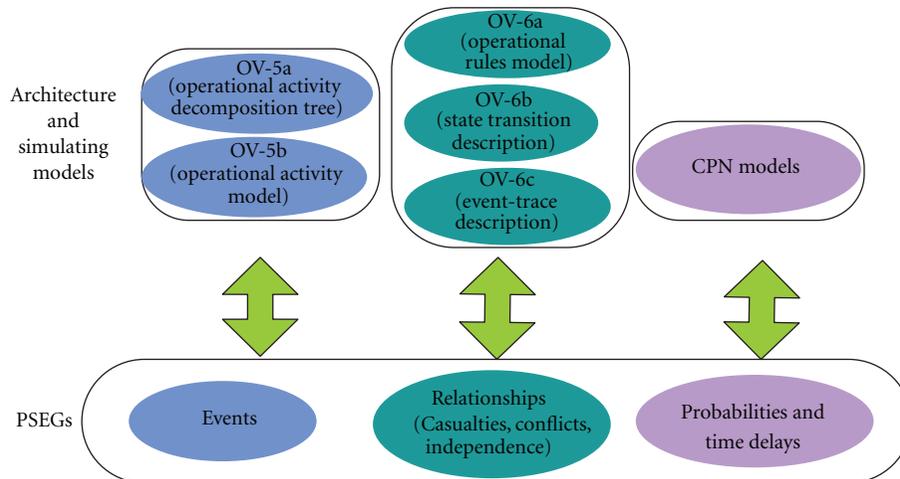


FIGURE 4: PSEGS translation rules.



FIGURE 5: Graphical forms of a minimal sequence and a minimal cycle event path.

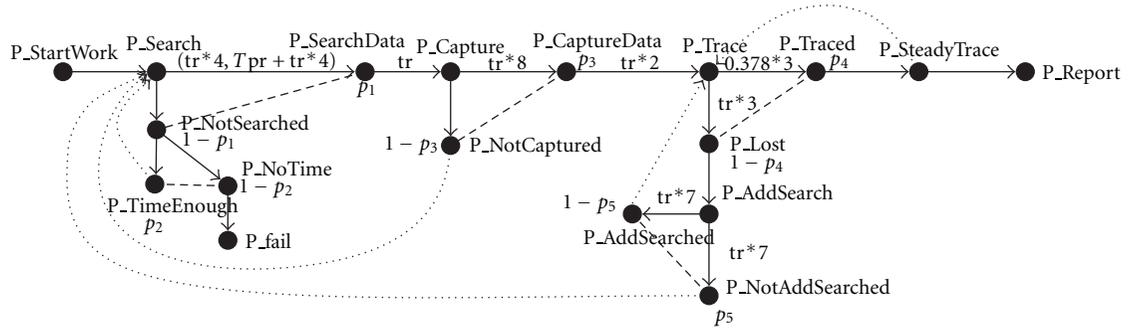


FIGURE 6: The PESG of the EWR system.

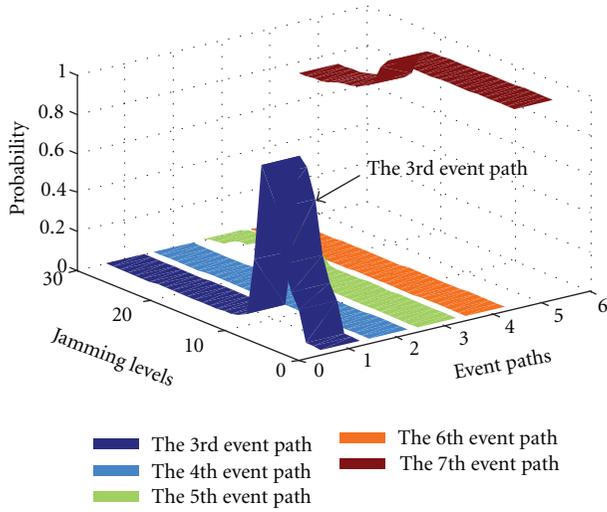


FIGURE 7: The changing currents of cycle event path probabilities under different jamming levels.

*Step 3.* Compute the probability and time delay of each event obtained from the simulation results of CPN models, and write them near the associated dots.

**3.2. Definitions and Analysis of Event Paths.** To analyze properties of every event path, we should firstly define the event paths, cycle, and sequence event paths according to relationships of events in PESGs. In this section, we give the following definitions where causalities are the basic relationships.

**Definition 3 (Event Paths).** For  $e_1, e_2 \in E$ , if  $e_1 \rightarrow e_2$ , then we call  $e_1 \rightarrow e_2$  a casual event pair. A serial casual event included

TABLE 2: Names and meanings of the EWR system events.

Names	Meanings
P_StartWork	EWR starts working.
P_Search	EWR starts to search.
P_SearchData	EWR obtains searching data.
P_NotSearched	EWR searches no threats.
P_TimeEnough	EWR works no overtime.
P_NoTime	EWR works overtime.
P_Fail	EWR fails.
P_Capture	EWR starts to capture.
P_CaptureData	EWR obtains searching data.
P_NotCaptured	EWR captures no threats.
P_Trace	EWR starts to track.
P_Lost	EWR loses threats.
P_AddSearch	EWR starts to add search.
P_AddSearched	EWR add-searches threats.
P_NotAddSearched	EWR add-searches no threats.
P_Traced	EWR tracks threats.
P_SteadyTrace	EWR tracks threats steadily.
P_Report	EWR transmit reports.

with multicausal event pairs is defined to be an event path, the first event  $e_0$  of which is called the initial event, and the last event  $e_n$  the final event.

**Definition 4 (Cycle Event paths).** If the final event  $e_n$  and the initial event  $e_0$  of one event path satisfy  $e_n \rightarrow e_0$ , that is to say, there exists at least one cycle event, then we define this event path to be a cycle event path. And we call it a minimal cycle event path if there is no other cycle event except  $e_n$ .

TABLE 3: Event paths and meanings of the EWR system.

Routess	Meanings
$\sigma_1 = P\_StartWork; P\_Search; P\_SearchData; P\_Capture; P\_CaptureData; P\_Trace; P\_Traced; P\_SteadyTrace; P\_Report$	EWR transmits reports successfully.
$\sigma_2 = P\_StartWork; P\_Search; P\_NotSearched; P\_NoTime; P\_Fail$	EWR works overtime and does not find any threats.
$\sigma_3 = P\_Search; P\_NotSearched; P\_TimeEnough$	EWR searches again when nothing found.
$\sigma_4 = P\_Search; P\_SearchData; P\_Capture; P\_NotCaptured$	EWR captures again when nothing captured.
$\sigma_5 = P\_Trace; P\_Lost; P\_AddSearch; P\_AddSearched$	EWR add-searches threats after lost them.
$\sigma_6 = P\_Search; P\_SearchData; P\_Capture; P\_CaptureData; P\_Trace; P\_Lost; P\_AddSearch; P\_NotAddSearched$	EWR searches again after not add searched.
$\sigma_7 = P\_Trace; P\_Traced; P\_SteadyTrace$	EWR tracks steadily.

*Definition 5* (Sequence Event Paths). If the initial event  $e_0$  and the final event  $e_n$  of one event path satisfy that  $\neg \exists e \in E$  such that  $e \mapsto e_0$  or  $e_n \mapsto e$ , then we define this event path to be a sequence event path. And we call it a minimal sequence event path if there is no other sequence event except  $e_n$ . Let  $\sigma$  be an event path, we denote  $\sigma[i] = e_i$  as the  $i$ th event of  $\sigma$ .

In PSEGs, the part of just concluded dots and arrows is a sequence event path, and it is a minimal sequence event route if there is only an initial event and a final event. And if an event path consists of dots, arrows, and only one dash-dotted arrow from the final event to the initial event, then it is a cycle event path. They are shown in Figure 5.

In recent literatures, [15, 16] a key event path is the node serials with the most sums of all node time delays, and the activities of the key event path are called key activities. Or a key event path is the transition serials with the least time delays, which is also called the main route. Then according to different requirements, a key event path can be represented as the one with the most or least time delays. From Definitions 3–5, the cycle event paths may affect the time delays of some sequence event paths. With the bigger probabilities of the cycle event paths, the time delays of associated sequence event paths will be more. Therefore, we mainly consider the probabilities of cycle event paths. In other sides, the disturbance analysis method is an efficient one to analyze discrete event systems, which we can adopt to compute the changing scope of each cycle event path, and the key event path is the one with the maximal changing probabilities.

The sensitive analysis method can be used to obtain the changing probabilities. Sensitivity is to focus on the grades of some system parameters changing with certain indexes. In this paper, we can find the key event path as the cycle event path with the most probability sensitivity.

#### 4. An Example

Take the CPN model of the early warning radar (EWR) system [17] as an example. The EWR system starts searching in the directed searching screen once it receives the leading information from the command and control system. Then it turns to the capturing state if threats found. The capturing state is a transitional process between searching and tracking, after which the EWR system starts tracking. During the

continuous tracking state, it computes the ballistic trajectory parameters of threats and transmit them to the command and control system.

According to the PSEGs translation rules and activities of OV-5b, we can extract all events of the EWR system as in Table 2.

According to the associated OV-6 models, we can finish the PSEG of the EWR system as in Figure 6, where  $tr$  is the tracking pulse cycle of EWR, and  $Tpr$  is the time delay for EWR scanning the whole search screen once.

There are 2 minimal sequence event paths ( $\sigma_1, \sigma_2$ ) and 5 minimal cycle event paths ( $\sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7$ ) in Figure 6. They are as Table 3.

From Figure 6, the relationships between each cycle event path probability and each conflict probability are as follows:

$$\begin{aligned}
 p_{\sigma_3} &= (1 - p_1)p_2, & p_{\sigma_4} &= p_1(1 - p_3), \\
 p_{\sigma_5} &= (1 - p_4)(1 - p_5), & & \\
 p_{\sigma_6} &= p_1p_3(1 - p_4)p_5, & p_{\sigma_7} &= p_4.
 \end{aligned} \tag{2}$$

Using disturbance simulation method, all cycle event path probabilities can be computed by the Data Collector of CPN Tools. The colored ribbon figure of  $p_{\sigma_3}, p_{\sigma_4}, p_{\sigma_5}, p_{\sigma_6}$ , and  $p_{\sigma_7}$  changing with jamming levels is as Figure 7. We can learn that the 3rd cycle event path is the key event path of the EWR system; that is, the searching segment is weaker and more important than other segments. In theory, whether the EWR system can find threats in the leading search screen in time has the most influence to early warning results.

#### 5. Conclusion

The key event path analysis is an important part of discrete event system structural analysis. We proposed a key event path analysis approach of integrated systems based on the analysis of cycle event paths in PSEGs. Besides that, we brought forward translation rules between two different architecture modeling methods. The approach supports to develop structural analysis of key event paths, and it is worthy to study deeply in the structural analysis of discrete event dynamic systems.

## Acknowledgment

This paper is supported by the National Natural Science Foundation of China (Grant no. 60874068).

## References

- [1] DoD Architecture Framework Working Group, *DoD Architecture Framework Version 2.0, Volume 1: Introduction, Overview, and Concepts*, The United States: Department of Defense, 2009.
- [2] DoD Architecture Framework Working Group, *DoD Architecture Framework Version 2.0, Volume 2: Architectural Data and Models*, The United States: Department of Defense, 2009.
- [3] DoD Architecture Framework Working Group, *DoD Architecture Framework Version 2.0, Volume 3: DoDAF Meta-Model*, The United States: Department of Defense, 2009.
- [4] F. Ni, M. Z. Wang, F. Zhou et al., "Integrated architecture modeling approach from DoDAF to HCPN," *Systems Engineering and Electronics*, vol. 32, pp. 959–965, 2010.
- [5] A. H. Levis and L. W. Wagenhals, "C4ISR architectures: I. Developing a process for C4ISR architecture design," *Systems Engineering*, vol. 3, no. 4, pp. 225–247, 2000.
- [6] L. W. Wagenhals, I. Shin, D. Kim, and A. H. Levis, "C4ISR architectures: II. A structured analysis approach for architecture design," *Systems Engineering*, vol. 3, no. 4, pp. 248–287, 2000.
- [7] K. Jensen, *Colored Petri Nets Basic Concepts, Analysis Methods and Practical Use*, vol. 1, Springer, Berlin, Germany, 2nd edition, 1992.
- [8] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, 2007.
- [9] M. A. Piera and G. Mušič, "Coloured Petri net scheduling models: timed state space exploration shortages," *Mathematics and Computers in Simulation*, vol. 82, pp. 428–441, 2010.
- [10] Z. Xiao and Z. Ming, "A method of workflow scheduling based on colored Petri nets," *Data and Knowledge Engineering*, vol. 70, no. 2, pp. 230–247, 2011.
- [11] B. Rachid and E. M. Abdellah, "On the analysis of some structural properties of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 35, no. 6, pp. 784–794, 2005.
- [12] V. Daniele and Y. Nobuko, "Probabilistic  $\pi$ -calculus and event structures," *Electronic Notes in Theoretical Computer Science*, vol. 190, no. 3, pp. 147–166, 2007.
- [13] R. Langerak, *Transformation and Semantics for LOTOS [Ph.D. thesis]*, University of Twente, 1992.
- [14] G. Winskel, "An introduction to event structures," *Lecture Notes in Computer Science*, vol. 354, pp. 364–397, 1989.
- [15] R. Liu, A. Kumar, and W. van der Aalst, "A formal modeling approach for supply chain event management," *Decision Support Systems*, vol. 43, no. 3, pp. 761–778, 2007.
- [16] E. Bland, S. Y. Brian, F. B. James, F. L. Robert, and D. F. Lisa, "Event-driven competing risks," *Ecological Modelling*, vol. 158, no. 1-2, pp. 35–50, 2002.
- [17] L. S. Melvin and P. B. Gerald, "Radars for the detection and tracking of ballistic missiles, satellites and planets," *Lincoln Laboratory*, vol. 12, pp. 217–243, 2000.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

