J Supercomput (2011) 56: 79–105
DOI 10.1007/s11227-009-0343-0

Modeling and survivability analysis of service composition using Stochastic Petri Nets

Yuanzhuo Wang · Chuang Lin ·
Peter D. Ungsunan · Xiaomeng Huang

Published online: 20 October 2009

Abstract In this paper, we propose a service composition model method that supports quantitative computation based on Stochastic Petri Nets (SPN). It can capture the semantics of complex service combinations and their respective specifications. In this method, services are divided into interior services and exterior services. The exterior services will be published to the users, while the interior ones do not need to be published. Six equivalent simplified theorems which can be used to simplify the complex models of interior services to simple models of exterior services are presented. They enable the minimization of the state space of the model and make quantitative computation feasible. In addition, since Grid services are always affected by all kinds of churns in actual applications, we also research survivability and its main attributes for Grid service composition. The definition and computational methods based on the model are put forward. In the end, we use the method presented above to describe and analyze an example of travel Grid services successfully.

Keywords Service composition · Grid · Survivability · Stochastic Petri Net · Fault tolerance

1 Introduction

Grids have emerged as a global cyber-infrastructure for the next-generation of science applications, by integrating large-scale, distributed and heterogeneous resources. Scientific communities, such as high-energy physics, gravitational-wave physics, geophysics, astronomy, and bioinformatics, are utilizing Grids to share, manage and process large data sets. In order to support complex scientific experiments, distributed

Y. Wang (✉) · C. Lin · P.D. Ungsunan · X. Huang
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
e-mail: wangyuanzhuo@tsinghua.edu.cn

resources such as computational devices, data, applications, and scientific infrastructures should be orchestrated through services [1]. These services, such as GridPhyN [3] and GridAnt [4], which are service-based, are based on the Open Grid Service Architecture (OGSA) [2]. OGSA integrates main Grid technologies with web services mechanisms to create a distributed system framework. It defines mechanisms for creating, managing, and exchanging information among entities called Grid services. Since Grid services are developed in virtual organizations worldwide, it is very difficult and costly to develop a suitable Grid service which satisfies the needs of every practical application. The significant potential of Grid services can be achieved through Grid service composition that produces more sophisticated functions compared to single Grid services. Grid service composition which combines some available services primarily addresses the situation of when a user's request cannot be satisfied by any available single service. The research of the Grid service composition model method can refer to traditional workflow, which consists of some fundamental components, such as the process model, running mechanism, user and application. In [5], the authors discuss the workflow modeling of several complex grid services, composed by simple ones, using a High-Level Petri Net. In [6], the authors, using the real case study of the Fraunhofer Resource Grid, use the a Petri Net-based graph model for orchestrating grid services and legacy command line applications, in order to model complex services by composing simple ones. However, there are still some distinctions between them. Traditional workflow emphasizes simplex directional information flow and control flow, but Grid service composition requires cooperation and interaction among different Grid services. On the other hand, there are some useful investigations of web service composition. Furthermore, the Grid service is similar to web services, but the Grid service has states. It is preferable that the numbers of Grid resources are adaptable to requirements of different complex tasks such as to provide felicitous and survivable service composition. At the same time, since Grid resources are distributed, heterogeneous and dynamic, the Grid service is more easily affected by accidental churns, such as attacks, failures, and accidents. These characteristics bring challenges to workflow and web service composition modeling and execution because there is a lack of dynamic modeling and corresponding scheduling. So it is necessary to research more appropriate model and analysis methods for development of Grid service composition. Based on the method, some abilities of the Grid service will be computed and predicted.

Stochastic Petri Net (SPN) models have been proved to be effective for describing prioritized, concurrent, asynchronous, stochastic and nondeterministic events. On the one hand, the underlying stochastic processes which are involved in SPN have evolved from homogeneous Markov processes. They have the approximate modeling ability with the other stochastic model. On the other hand, SPN has the trait of graphical representation. It is visual, intuitive and extensible. However, a Markov model is always hard to understand to an ordinary user, and when some changes take place, it will be difficult for new components to be appended in the Markov model. At the same time, SPN has reached a degree of maturity such that it can handle realistically the current complex systems; the corresponding models can be built and processed under less stronger assumptions, and powerful software packages are available to assist the modeler, such as ESPN, GreatSPN, SPNP, SURF-2, TOMSPIN, and UltraSAN, etc. [7].

In this paper, we propose a service composition model and analysis method based on Stochastic Petri Net, which is composed of an interior service and exterior service. The exterior service will be published to the users. The interior one need not be published. The model is expressive enough to capture the semantics of complex service combinations and their respective specificities. The obtained framework enables declarative composition of Grid services. It shows dynamic and transient relationships among services. Then we propose the definition of the survivability of the Grid service composition, and introduce the survivability analysis method based on the Stochastic Petri Net model of the service composition. Through an example of travel Grid services, we demonstrate the validity of this method.

The remainder of the paper is organized as follows. The related work is argued in Sect. 2. Section 3 presents the overview of the service composition and the service survivability definition and attributes. Section 4 introduces a new Grid service composition model based on Stochastic Petri Net. Section 5 investigates the method of the survivability analysis for Grid service composition. In Sect. 6, an application of Grid service composition is introduced. Section 6 concludes the paper and identifies the future work.

2 Related work

Many standards and languages based on web services have been proposed, such as WSFL [8] and BPEL [9]. There also are workflow model oriented services, such as GSFL [10, 11]. In [12], models and formalisms for e-services are proposed from the perspectives of workflow, process models, and automata theory. Raman et al. [13] developed a comprehensive architecture for the creation, placement, and management of services composition. Semantically described services can make it possible to improve the precision of the search for existing services and to automate parts of the service composition process. OWL-S [14] and WSML [15] have been proposed as competing semantic web service languages to address the need for semantically defined services. Gronmo and Jaeger [16] proposed that semantic web service languages can be utilized within a model-driven methodology for building composite web services. In [17], an abstract framework, called Colombo, is presented, where services are characterized in terms of message exchanges, data flow and effects on the real world. In [18], an approach models services using Petri nets, and constructs a service net with input and output places corresponding to a service's initial and final states. Motahari-Nezhad et al. [19] argue that solution-reuse at a large scale can be exploited to address challenges of service composition and integration by harnessing the collective intelligence and labor of various businesses and people present on the Internet. These models and describing languages focus on what a service or composition does, either in terms of the input/output of services and their impact on the world, or the sequencing of the activities in a service. They contain a detailed description of the inputs and outputs in the services, from which the users know how to use these services. But they cannot provide an exact quantitative analysis of the capability to deliver essential services for users' tasks in the presence of attacks, failures, or accidents. For users, it is hard to know the quality of outputs even if the certain inputs are given. For service managers, it is difficult to design the processes effectively. It also brings a lot of problems during process tracing and monitoring. We need some visual modeling techniques to model and analyze the Grid service ahead. Visual representations can provide a high-level yet precise language which allows it to express and figure out the concepts at their natural level of abstraction. In [20] the authors model services as Petri nets [21] by assigning transitions to methods and places to states. Each service has an associated Petri net that describes service behavior and has two ports: one input place and one output place. But they neglect the stochastic specialty of Grid services. A Grid service behavior is basically a partially ordered set of operations and has some stochastic traits.

3 Service composition

3.1 Overview of service composition

In this paper, we have designed a service composition model method to adapt to the requests of Grid services. In this model, whether a service is a simple service or a complex one depends on the request of the user task. Simple services have special functions. They can be combined together to build up a complex service.

In general, services can be divided into two types: interior services and exterior services. An interior service contains all implementation information that the service managers need, and it will not be published to users. An exterior service can be obtained by simplifying an interior service. Compared with interior services, the structure of the exterior services may be simpler. An exterior service is published to the users to tell them what this service can do and what the inputs and outputs are. The published part also includes the mapping relationship between the inputs and outputs of the service, from which we can trace and monitor the process state easily and a process designer can design a process effectively. This kind of service can be considered as half-black-box. Since interior services will never be published generally, users will not know the detailed implementation information of this service, which also satisfies the security requirements. When the implementation process of a service is changed, it is not necessary to tell the user what has been changed unless the exterior service is also changed.

3.2 Survivability of service composition

In Grid environments, there are a large number of similar or equivalent resources provided by third parties. According to users' tasks, the Grid service manager needs to select suitable resources for material service composition applications. The services which provide the same function may have different survivability. The survivability is a capability to deliver essential services to users' tasks in the presence of attacks, failures, or accidents. Different users or applications may have different expectations and requirements. Therefore, it is not sufficient to only consider functional characteristics of the service composition when distributing tasks. For different users' QoS requirements, such as time limit (deadline) and expenditure limit (budget), there are different

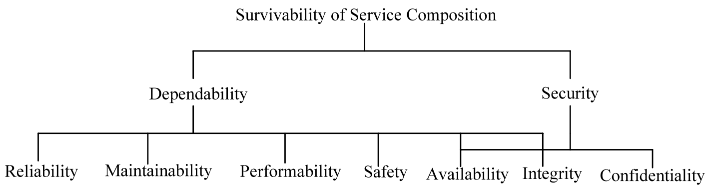


Fig. 1 Survivability of service composition

service survivability measures. Users must be able to specify their QoS expectations of the tasks on the task scheduling level. Then, actions conducted by Grid service managers using run-time must be picked out according to initial service survivability measures.

Figure 1 shows the main attributes of survivability for Grid service. It includes two dimensions: dependability and security. Dependability is related to probability of failures and repairs for the task execution. As developed over the past three decades, dependability is an integrating concept that encompasses the following attributes: maintainability, reliability, safety, performability, availability and integrity [20]. Security refers to confidentiality of the execution of tasks and the trustworthiness of resources. Generally security embodies such attributes as availability, integrity, confidentiality [20].

3.3 Mathematics description

In this part, we will briefly introduce the main features of SPN and the mathematical description of some fundamental concepts we will use, so that the model presented in this paper can be better understood. Readers are assumed to have some basic knowledge of Petri nets as mentioned in [23].

A Petri net is a graphical model useful for modeling systems exhibiting concurrent, asynchronous nondeterministic behaviors [21]. SPN is Petri net augmented with a set of average transition rates for the exponentially distributed transition-firing times. A transition represents a class of possible marking changes. Such a change, also called transition firing, may be induced by removing tokens from the input places of the transition or adding tokens to the output places of the transition according to the expressions labeled on the arcs. A transition may be associated with an enabling predicate which can be expressed in terms of the place marking expressions. If the predicate of a transition is false, the transition will be disabled. In SPN models, transitions can be categorized into two classes: transitions of Class 1 represent logical relations or determine if some conditions are satisfied [24]. This class of transitions is called immediate transition with zero firing time. Transitions of Class 2 represent the operations on the tasks or information processing. This class of transitions is called timed transition with exponential distributed firing time. A marking in an SPN model represents a distribution of tokens in the model. The state space of a model consists of the set of all markings reachable from the initial marking through the occurrence of transition firing. An SPN is homomorphic to a continuous-time Markov Chain (MC), and there is a one-to-one relationship between markings of the SPN and states of the MC [24, 25].

Definition 1 (Stochastic Petri Net) A Stochastic Petri Net (SPN) is a quadruple (P, T, F, λ), where

P is a finite set of place;
T is a finite set of transitions ($P \cap T \neq \Phi$);
$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs;
$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ is a set of firing rates of transitions in transition set.

A Grid service behavior is basically a partially ordered set of operations. And using SPN, these operations can be modeled by transitions and the state of the service is modeled by places. The arcs between places and transitions are used to specify causal relations. The SPN model for a Grid service needs one input place for absorbing information, and one output place for emitting information; it will facilitate the definition of the composition operators and the analysis. At any given time, when a token is in the input place, this means that corresponding Grid service is in the Ready state, whereas a token in the output place, it means that the Grid service is in the Completed state.

Definition 2 (Service Composition Net) A service net is a tuple $SCN = (P; T; F; \lambda; i; o)$, where (P, T, F, λ) is an SPN, and

i is the input place with $i = \{x \in P \cup T | (x, i) \in F\} = \phi$;
o is the output place with $o = \{x \in P \cup T | (o, x) \in F\} = \phi$;
Place i is considered as the initial marking of a service S (i.e., only i contains a token). The execution of S starts when a token is in place i and terminates when a token reaches place o.

Definition 3 (Survivability of Grid Service) The survivability of Grid service is a tuple *Surv* = (*Reliability*, *MTTF*, *Maintainability*, *MTTR*, *Availability*, *Safety*, *Performability*, *Integrity*, *Confidentiality*, ...), where

Reliability: continuity of correct service. It can be computed as $R(t) = P\{X > t\}$.

MTTF: Mean Time to Failure; in some research, people also use it to describe the reliability of a system [26]. It is always denoted by $\mathrm{E}[\tau]$.

Maintainability: ability to undergo modifications and repairs [22]. It can be computed as $M(t) = P\{X \leq t\}$.

MTTR: Mean Time to Repair. It often denotes the average time to repair Grid nodes. Sometimes people also use it to describe the maintainability of a system. It can be shown by $WTTR/n$, where *WTTR* is the whole time to repair and n is times of repair.

Availability: readiness for correct service [22]. Steady state availability is expressed as the fraction of time that a system can be used for its intended purpose during a specified interval of time or in steady state. The fraction of time is a time interval: $A_s = \lim_{t\to\infty} A_I(t) = \lim_{t\to\infty}(\int_0^t A_I(u)du/t)$, where $A_I(t)$ is the probability that the system is properly running on time.

Safety: absence of catastrophic consequences on the users and the environment. It can be computed as $S = \sum_{i=S_W} \pi_i$, where S_W is the set of states under no destructive failure, π_i is the steady state probability.

Performability: quantifies system performance in the presence of failures. It can be computed as $P(L,t)$, which denotes the probability that the performance of system is L at t.

Integrity: absence of improper system alterations. It can be computed as $I = \sum_{i=S_i} \pi_i$, where S_I is the set of states with no improper system alterations, π_i is the steady state probability.

Confidentiality: the absence of unauthorized disclosure of information. $C = \sum_{i=S_C} \pi_i$, where S_C is the set of states with no unauthorized disclosure of information, π_i is the steady state probability.

Definition 4 (Grid Service) A Grid service is a tuple $GS = (TaskDe; SerDe; ResDe; SCN; Surv)$, where

TaskDe is a description of the user QoS requests, such as deadline and budget.
SerDe = (*SerID*, *CS*) is a description of the combinatorial service, including *SerID* and *CS*, where *SerID* is the name of the service, used as its unique identifier, *CS* is a set of its component services.
$SCN = (P; T; F; \lambda, i; o)$ is the service net modeling the dynamic behavior of the service.
$ResDe = (NodeID, URL, P, T, C)$ is the description of resources provided by servers, including *NodeID*, *URL*, P, R, where *NodeID* is the resource name, *URL* is the orientation of the resource in the network, P represents the price of the resource, and T denotes the anticipative time completing task successfully. For a complicated task, it refers to the total time required for completing the execution of a group of services. C represents the cost associated with services execution including the cost for the resource management system and usage charge of Grid resources for processing tasks.
Surv = (*Reliability*, *MTTF*, *Maintainability*, *Availability*, *MTTR*, *Safety*, *Performability*, *Integrity*, *Confidentiality*, ...) is the description of the capability to deliver essential services to users in the presence of attacks, failures, or accidents.

When the parameters mentioned above about user QoS requests and resource descriptions are certain, the survivability *Surv* and its attributes will be computed according to their definitions and relational formulae.

4 Service composition model

In heterogeneous and dynamic distributed systems like the Grid, detailed monitoring of workload and its resulting system configuration are required to facilitate quality of service diagnosis and adaptive performance tuning [27]. In this part, we will describe the main service composition patterns by the algebra operators. These patterns are chosen to allow basic and advanced Grid service compositions. The set of services can be defined by the following:

$$GSF = \{\varepsilon | X | S_1 \rightarrow S_2 | S_1 \| S_2 | S_1 \varPi S_2 | \mu S | S_1 \alpha \| \beta S_2 | (k/n)(S_1, S_2, \ldots, S_n)\}$$

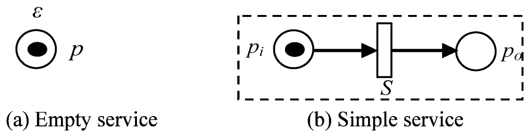


Fig. 2 Empty service and simple service

where *GSF* denotes the set of services and the elements of the set are possible service patterns. These symbols of service patterns will be introduced in detail in the following sections.

4.1 Basic pattern

In this section, we will introduce the modeling method of the basic patterns in detail, including the empty service, simple service, sequential service, parallel service, choice service and iterative service. At the same time, the equivalent simplification theorems are given. The corresponding proofs have been given in [23].

Empty Service: The empty service ε is a service that performs no operation. It depicts the situation when no tasks are operated or the service fails.

Definition 5 The empty service ε is defined as $\varepsilon = (TaskDe; SerDe; SCN; ResDe; Surv)$, where

$TaskDe =$ 'Empty';
$SerDe =$ (Empty; Empty);
$SCN = (\{p\}; \phi; \phi; \phi, p; p)$;
$ResDe =$ (Null, Null, Null, Null, Null);
$Surv =$ (Null, Null, Null, Null, Null, Null, Null, Null, ...).

The empty service can be shown by SPN in Fig. 2(a), which includes only one place.

Simple Service: The simple service X represents an atomic or basic service in this context.

Definition 6 The simple service X is defined as $X = (TaskDe; SerDe; SCN; ResDe; Surv)$, where some parameters, such as *TaskDe*, *ResDe*, *Surv*, can be certain when the service is used in the actual case.

$SerDe =$ (Simple Service, S)
$SCN = (P; T; F; \lambda, i; o)$, where
– $P = \{p_i, p_o\}$;
– $T = \{S\}$;
– $F = \{(p_i, S), (S, p_o)\}$;
– $\lambda = \{\lambda_1\}$.

Sequential Service: The sequential service $S_1 \to S_2 \cdots \to S_n$ represents a composite service that service S_1 followed by the service S_2, and S_n is the last one. $\to$ is a sequence operator.

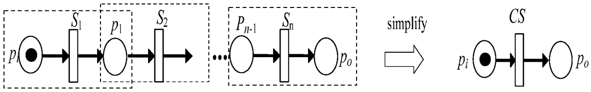


Fig. 3 Sequential service and its equivalent model

Definition 7 The sequential service *SS* is defined as *SS* = (*TaskDe*; *SerDe*; *SCN*; *ResDe*; *Surv*), where some parameters, such as *TaskDe*, *ResDe*, *Surv*, will be certain when the service is used in the actual case.

$SerDe = \{\text{Sequential Service}, \bigcup_{i=1}^{n} S_i\}$
$SCN = (P; T; F; \lambda, i; o)$, where
- $P = \{p_i, p_1, p_2, \ldots, p_{n-1}, p_o\}$;
- $T = \{S_1, S_2, \ldots, S_n\}$;
- $F = \{(p_i, S_1), (S_1, p_1), (p_1, S_2), \ldots, (p_{n-1}, S_n)(S_n, p_o)\}$;
- $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$.

The sequential service can be shown by SPN in Fig. 3.

Theorem 1 *Sequential service has n interior services,* $S_1, S_2, \ldots, S_n$*, which are denoted by n transitions. The service rate of each transition* S_i $(1 \le i \le n)$ *is an exponentially distributed random variable with expected value* λ_i*. Then the equivalent exterior service rate of the transition CS has been expected as* $\lambda = 1/(\sum_{i=1}^{n} 1/\lambda_i)$.

Parallel Service: The parallel service $S_1 \| S_2 \ldots \| S_n$ represents a composite service that performs all the services $S_1, S_2 \ldots S_n$ at the same time. $\|$ is a parallel operator.

Definition 8 The parallel service *PS* is defined as *PS* = (*TaskDe*; *SerDe*; *SCN*; *ResDe*; *Surv*), where some parameters, such as *TaskDe*, *ResDe*, *Surv*, will be certain when the service is used in the actual case.

$SerDe = \{\text{Parallel Service}, \bigcup_{i=1}^{n} S_i\}$
$SCN = (P; T; F; \lambda, i; o)$, where
- $P = \{p_i, p_{11}, p_{12}, p_{21}, p_{22}, \ldots, p_{n1}, p_{n2}, p_o\}$;
- $T = \{T_1, S_1, S_2, \ldots, S_n, T_2\}$;
- $F = \{(p_i, T_1), (T_1, p_{11}), (T_1, p_{21}), \ldots, (T_1, p_{n1}); (p_{11}, S_1), (S_1, p_{12}), (p_{21}, S_2), (S_2, p_{22}), \ldots, (p_{n1}, S_n), (S_n, p_{n2}); (p_{12}, T_2), (p_{22}, T_2), \ldots, (p_{n2}, T_2), (T_2, p_o)\}$
- $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$.

The parallel service can be shown by SPN in Fig. 4.

Theorem 2 *Parallel service has n interior services,* $S_1, S_2, \ldots, S_n$*, which are denoted by n transitions. The service rate of each transition* S_i $(1 \le i \le n)$ *is an exponentially distributed random variable with expected value* λ_i*. Then the equivalent*

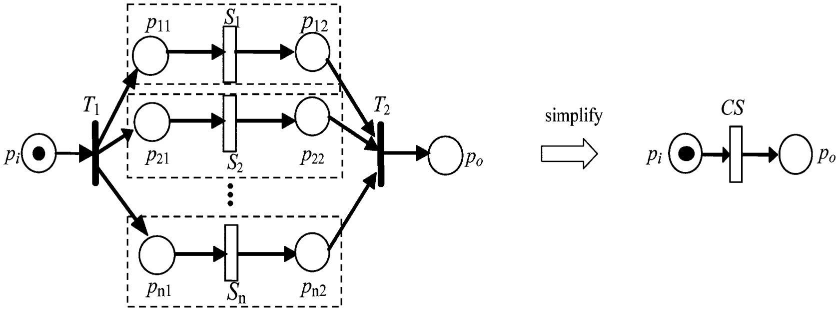


Fig. 4 Parallel service and its equivalent model

exterior service rate of the transition CS has been expected as

$$\lambda = 1\Big/\left(\sum_{i=1}^{n} 1/\lambda_i - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 1/(\lambda_i+\lambda_j) + \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=j+1}^{n} 1/(\lambda_i+\lambda_j+\lambda_k) + \cdots + (-1)^{n-1}1\Big/\sum_{i=1}^{n}\lambda_i\right).$$

Choice Service: The choice service $S_1 \prod S_2 \ldots \prod S_n$ represents a composite service that behaves as either service S_1 or service $S_2 \ldots$ or service S_n. Once one of them executes its operation, the other services are discarded. $\prod$ is a choice operator.

Definition 9 The choice service *CHS* is defined as *CHS* = (*TaskDe*; *SerDe*; *SCN*; *ResDe*; *Surv*), where some parameters, such as *TaskDe*, *ResDe*, *Surv*, will be certain when the service is used in the actual case.

SerDe = {Choice Service, $\bigcup_{i=1}^{n} S_i$}
SCN = $(P; T; F; \lambda, i; o)$, where
- $P = \{p_i, p_o\}$;
- $T = \{S_1, S_2, \ldots, S_n\}$;
- $F = \{(p_i, S_1), (p_i, S_2), \ldots, (p_i, S_n); (S_1, p_o), (S_2, p_o), \ldots, (S_n, p_o)\}$;
- $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$.

The choice service can be shown by SPN in Fig. 5.

Theorem 3 *Choice service has n interior services, $S_1, S_2, \ldots, S_n$, which are denoted by n transitions. The service rate of each transition S_i $(1 \le i \le n)$ is an exponentially distributed random variable with expected value λ_i. Transition S_i can fire with probability α_i, and $\sum_{i=1}^{n} \alpha_i = 1$. Then the equivalent exterior service rate of the transition CS has been expected as $\lambda = 1/\sum_{i=1}^{n} \alpha_i/\lambda_i$.*

Iterative Service: The iterative service μS represents a service that performs a certain number of times the service S. μ represents an iteration operator.

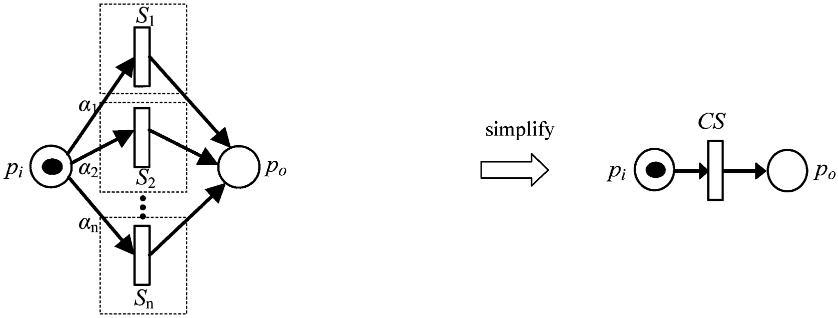


Fig. 5 Choice service and its equivalent model

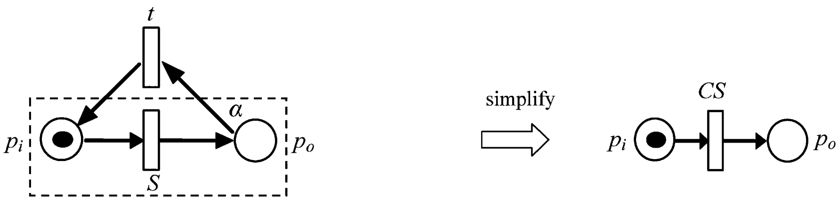


Fig. 6 Iterative service and its equivalent model

Definition 10 The iterative service *IS* is defined as *IS* = (*TaskDe*; *SerDe*; *ResDe*; *SCN*; *Surv*), where some parameters, such as *TaskDe*, *ResDe*, *Surv*, will be certain when the service is used in the actual case.

SerDe = {Iterative Service, S}
SCN = $(P; T; F; \lambda, i; o)$, where
- $P = \{p_i, p_o\}$;
- $T = \{S, t\}$;
- $F = \{(p_i, S), (S, p_o), (p_o, t), (t, p_i)\}$;
- $\lambda = \{\lambda_1, \lambda_2\}$.

The iterative service can be shown by SPN in Fig. 6.

Theorem 4 *Iterative service has an interior service S and a delay t. There are two transitions, S and t. The service rate of transition S and delay rate of transition t are exponentially distributed random variables with expected values λ_1 and λ_2. The transition t can fire with probability α. Then the equivalent exterior service rate of the transition CS has been expected as $\lambda = (\alpha\lambda_1 + \lambda_2)/\lambda_1\lambda_2(1-\alpha)$.*

4.2 Advanced pattern

Based on the basic Grid service composition patterns mentioned above, in this section, some advanced Grid service composition patterns are researched and corresponding equivalent simplification theorems are proved.

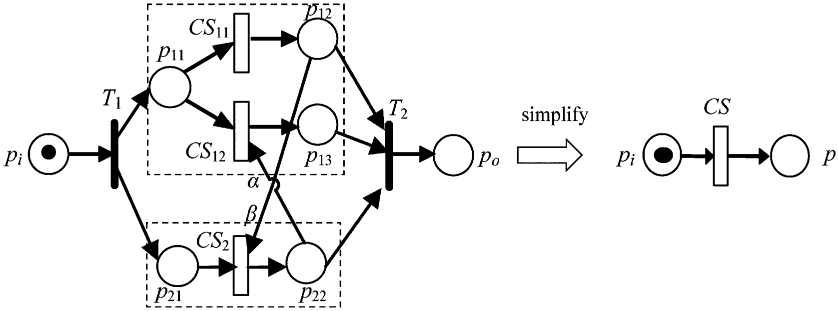


Fig. 7 Dependent service and its equivalent model

Dependent Service: The dependent service $CS_1\alpha\|\beta CS_2$ represents a composite service that performs the services CS_1 and CS_2, and they are dependent on each other. α and β denote these two composite service-dependent probabilities. $\|$ is a parallel operator.

Definition 11 The dependent service *DS* is defined as $DS = (TaskDe; SerDe; SCN; ResDe; Surv)$, where some parameters, such as *TaskDe*, *ResDe*, *Surv*, will be certain when the service is used in the actual case.

$SerDe = \{$Dependent Service; $CS_1 \cup CS_2\}$
$SCN = (P; T; F; \lambda, i; o)$, where
- $P = \{p_i, p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_o\}$;
- $T = \{T_1, CS_{11}, CS_{12}, CS_2, T_2\}$;
- $F = \{(p_i, T_1), (T_1, p_{11}), (T_1, p_{21}), (p_{11}, CS_{11}), (p_{11}, CS_{12}), (CS_{11}, p_{12}), (CS_{12}, p_{13}), (p_{21}, CS_2), (CS_2, p_{22}), (p_{12}, CS_2), (p_{22}, CS_{12}), (p_{12}, T_2), (p_{13}, T_2), (p_{22}, T_2), (T_2, p_o)\}$;
- $\lambda = \{\lambda_{11}, \lambda_{12}, \lambda_2\}$.

The dependent service can be shown by SPN in Fig. 7.

Theorem 5 *Dependent service has two composite services, CS_1 and CS_2, and they are dependent on each other. α and β denote these two composite service-dependent probabilities. CS_1 can be divided into two parts, CS_{11} and CS_{12}, according to their relationship with CS_2. There are three transitions, CS_{11}, CS_{12} and CS_2. The service rates of the transitions are exponentially distributed random variables with expected values $\lambda_{11}, \lambda_{12}$ and λ_2. The firing of transition CS_{12} needs CS_2 with probability α and the firing of transition CS_2 needs CS_{12} with probability β. Then the equivalent exterior service rate of the transition CS has been expected as*

$$\lambda = 1/(1/\lambda_{11} + 1/\lambda_{12} + 1/\lambda_2) - ((1-\alpha)\beta/(\lambda_{112}^{\rightarrow} + \lambda_{12}) + \alpha(1-\beta)/(\lambda_{122}^{\rightarrow} + \lambda_{11}) + (1-\alpha)(1-\beta)/(\lambda_{112}^{\|} + \lambda_{12}))$$

where, according to Theorems 1 and 2,

$$\lambda_{112}^{\rightarrow} = \lambda_{11}\lambda_2/(\lambda_{11}+\lambda_2),$$
$$\lambda_{122}^{\rightarrow} = \lambda_{12}\lambda_2/(\lambda_{12}+\lambda_2),$$
$$\lambda_{112}^{\parallel} = \lambda_{11}\lambda_2(\lambda_{11}+\lambda_2)/((\lambda_{11}+\lambda_2)^2-\lambda_{11}\lambda_2).$$

Proof Let the firing rate of transitions CS_{11}, CS_{12} and CS_2 be mutually independent, exponentially distributed.

According to the above description, we think the process can be divided into four instances: $S_{11} \to S_2 \to S_{12}$, $S_{11} \to S_2 \| S_{12}$, $S_{11} \| (S_2 \to S_{12})$ and $S_{11} \| S_2 \| S_{12}$. The response time of each instance can be expressed by the following formulae:

$$t_1 = 1/\lambda_{11} + 1/\lambda_2 + 1/\lambda_{12}, \tag{1}$$
$$t_2 = 1/\lambda_{11} + 1/\lambda_2 + 1/\lambda_{12} - 1/(\lambda_{112}^{\rightarrow} + \lambda_{12}), \tag{2}$$
$$t_3 = 1/\lambda_{11} + 1/\lambda_2 + 1/\lambda_{12} - 1/(\lambda_{11} + \lambda_{212}^{\rightarrow}), \tag{3}$$
$$t_4 = 1/\lambda_{11} + 1/\lambda_2 + \lambda_{12} - \big(1/(\lambda_{11}+\lambda_2) + 1/\big(\lambda_{112}^{\parallel} + \lambda_{12}\big)\big). \tag{4}$$

The probabilities that they happen are

$$P(t_1) = \alpha\beta, \tag{5}$$
$$P(t_2) = (1-\alpha)\beta, \tag{6}$$
$$P(t_3) = \alpha(1-\beta), \tag{7}$$
$$P(t_4) = (1-\alpha)(1-\beta). \tag{8}$$

So, the average response time is

$$t = \sum_{i=1}^{4} t_i \times P(t_i) = (1/\lambda_{11} + 1/\lambda_{12} + 1/\lambda_2) - \big((1-\alpha)\beta/(\lambda_{112}^{\rightarrow} + \lambda_{12}) + \alpha(1-\beta)/(\lambda_{122}^{\rightarrow} + \lambda_{11}) + (1-\alpha)(1-\beta)/\big(\lambda_{112}^{\parallel} + \lambda_{12}\big)\big). \tag{9}$$

And then the equivalent exterior service rate of the transition CS has been expected as

$$\lambda = 1/(1/\lambda_{11} + 1/\lambda_{12} + 1/\lambda_2) - ((1-\alpha)\beta/(\lambda_{112}^{\rightarrow} + \lambda_{12}) + \alpha(1-\beta)/(\lambda_{122}^{\rightarrow} + \lambda_{11}) + (1-\alpha)(1-\beta)/(\lambda_{112}^{\parallel} + \lambda_{12})).$$

Based on the above proofs, Theorem 5 is confirmed. □

Vote Service: The vote service $(k/n)(S_1, S_2, \ldots, S_n)$ represents a composite service that completes k services of all n services, then the task can be executed successfully, i.e., (k/n) is a vote operator.

Definition 12 The vote service *VS* is defined as *VS* = (*TaskDe*; *SerDe*; *SCN*; *ResDe*; *Surv*), where some parameters, such as *TaskDe*, *Surv*, *ResDe*, will be certain when the service is used in the actual case.

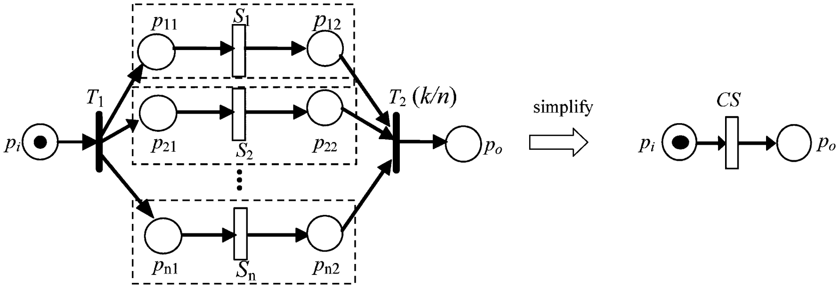


Fig. 8 Vote service and its equivalent model

$SerDe = \{\text{Vote Service}; \bigcup_{i=1}^{n} S_i\}$
$SCN = (P; T; F; \lambda, i; o)$, where

- $P = \{p_i, p_{11}, p_{21}, \ldots, p_{n1}, p_{12}, p_{22}, \ldots, p_{n2}, p_o\}$;
- $T = \{T_1, S_1, S_2, \ldots, S_n, T_2\}$;
- $F = \{(p_i, T_1), (T_1, p_{11}), (T_1, p_{21}), \ldots, (T_1, p_{n1}); (p_{11}, S_1), (S_1, p_{12}), (p_{21}, S_2), (S_2, p_{22}), \ldots, (p_{n1}, S_n), (S_n, p_{n2}); (p_{12}, T_2), (p_{22}, T_2), \ldots, (p_{n2}, T_2), (T_2, p_o)\}$
- $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$.

Theorem 6 *Vote service has n interior services, $S_1, S_2, \ldots, S_n$, which are denoted by n transitions. The service rate of each transition S_i $(1 \le i \le n)$ is an exponentially distributed random variable with expected value λ_i. Then, when $\lambda_1 = \lambda_2 = \cdots = \lambda_n$ and $\lambda_i = \lambda_0$, the equivalent exterior service rate of the transition CS has been expected as $\lambda = 1/\sum_{i=k}^{n} 1/i\lambda_0$.*

Proof Let the firing rate of transitions $S_1, S_2, \ldots, S_n$ be mutually independent, exponentially distributed. We assume the execution probability of service S_i is q_i and the undone probability within t time is p_i, so $p_i + q_i = 1$. The service rate of each service is an exponentially distributed random variable with expected value λ_i, then $p_i = e^{-\lambda_i t}$. When $\lambda_1 = \lambda_2 = \cdots = \lambda_n$ and $\lambda_i = \lambda_0$, then $p_i = p = e^{-\lambda_0 t}$. We can obtain the execution probability within t time

$$P_{k/n}(t) = \sum_{i=k}^{n} C_n^i p^i (1-p)^{n-i} = \sum_{i=k}^{n} C_n^i e^{-i\lambda_0 t}\left(1 - e^{-\lambda_0 t}\right)^{n-i}. \tag{10}$$

Then

$$\begin{aligned} \frac{1}{\lambda} &= \int_0^\infty P_{k/n}(t)dt = \int_0^\infty \left[\sum_{i=k}^{n} C_n^i e^{-i\lambda_0 t}\left(1 - e^{-\lambda_0 t}\right)^{n-i}\right]dt \\ &= \sum_{i=k}^{n} C_n^i \int_0^\infty e^{-i\lambda_0 t}\left(1 - e^{-\lambda_0 t}\right)^{n-i} dt = \sum_{i=k}^{n} \frac{1}{i\lambda_0}. \end{aligned} \tag{11}$$

We only need to prove

$$C_n^i \int_0^\infty e^{-i\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-i} dt = 1/i\lambda_0. \tag{12}$$

The mathematical induction proves as follows:

Firstly let $i = 1$,

$$\begin{aligned} C_n^1 \int_0^\infty e^{-\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-1} dt = 1/\lambda_0 = (n/\lambda_0) \int_0^\infty \left(1-e^{-\lambda_0 t}\right)^{n-1} d\left(1-e^{-\lambda_0 t}\right) \\ = (n/\lambda_0)(1/n)\left(1-e^{-\lambda_0 t}\right)\big|_0^\infty = 1/\lambda_0. \end{aligned} \tag{13}$$

Secondly, we assume $i = k\ (1 \le k < n)$, then (13) is confirmed:

$$C_n^k \int_0^\infty e^{-k\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-k} dt = 1/k\lambda_0. \tag{14}$$

Then

$$\begin{aligned} &C_n^{k+1} \int_0^\infty e^{-(k+1)\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-(k+1)} dt \\ &= \left(C_n^{k+1}/\lambda_0(n-k)\right) \int_0^\infty e^{-k\lambda_0 t} d\left(1-e^{-\lambda_0 t}\right)^{n-k} \\ &= \left(C_n^{k+1}/\lambda_0(n-k)\right)\left\{e^{-k\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-k}\big|_0^\infty - \int_0^\infty \left(1-e^{-\lambda_0 t}\right)^{n-k} d\left(e^{-k\lambda_0 t}\right)\right\} \\ &= \left(C_n^{k+1}/\lambda_0(n-k)\right)\left\{-\int_0^\infty \left(1-e^{-\lambda_0 t}\right)^{n-k} d\left(e^{-k\lambda_0 t}\right)\right\} \\ &= \left(C_n^{k+1}/\lambda_0(n-k)\right)(n-k/k+1)C_n^{k+1}\left\{k\lambda_0 \int_0^\infty e^{-k\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-k} dt\right\}. \end{aligned} \tag{15}$$

Bringing the above result equation (15) into (14), we can obtain

$$C_n^{k+1} \int_0^\infty e^{-(k+1)\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-(k+1)} dt = 1/(k+1)\lambda_0. \tag{16}$$

It shows that if $i = k\ (1 \le k < n)$, equation (12) is confirmed, then when $i = k+1$, equation (13) is still confirmed. So

$$\frac{1}{\lambda} = \int_0^\infty \left[\sum_{i=k}^n C_n^i e^{-i\lambda_0 t}\left(1-e^{-\lambda_0 t}\right)^{n-i}\right] dt = \sum_{i=k}^n 1/i\lambda_0 \tag{17}$$

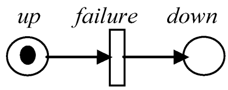


Fig. 9 Pure structure of the service failure model

and

$$\lambda = 1 \Big/ \sum_{i=k}^{n} 1/i\lambda_0. \tag{18}$$

Based on the above proofs, Theorem 6 is confirmed. □

For a complex service, we can consider the corresponding *GS* model as a hierarchy model. The whole *GS* model can be regarded as a model that is composed of some sub-models. Using the equivalent simplification methods, the sub-models will be equal to a time transition. Repeating these operations, complex Grid service composition can be simplified.

5 Survivability analysis of service composition

The SPN model mentioned above can describe the structure of the service composition and the relationship of each service. But it cannot show the effects to the Grid service when attacks, failures, or accidents happen. In this section, we will research the method of survivability analysis based on the definitions of the survivability and SPN.

5.1 Service failure model and repairable model

It is well known that the service failures may be caused by many accidents, such as blackouts, network attacks, operational accidents, service deadlocks, etc. Sometimes effects are easily measured, but their causes are not. We can get the service failure rate by historical statistical data. Many incidents can bring the same effect, so according to effects, what happened can be ignored when we model and evaluate the service composition. Here, the service failure model is very useful. As deterministic failure rate models are Poisson processes [28], the corresponding service failure models are multi-stage homogeneous Markov chains. A pure service failure model can be described using SPN, as in Fig. 9. It denotes a change of service status from normal to failure.

Repair is one of the main methods to resume the services, after the services fail. Each failure service has a failure-resumption process. And the relation between failure and resumption comprises two situations: critical systems and non-critical systems. *Critical systems*: the system can be repaired after each failure, and only if the system is repaired, the service ability can be resumed. *Non-critical systems*: the relation between failure and resumption is uncertain; it can be clarified by a stochastic process. Given a repairable model, we set two places to denote a system in an up state or a failure state, respectively. In order to indicate the failure process and repair process, we can describe the failure action (or repair action) by a transition failure

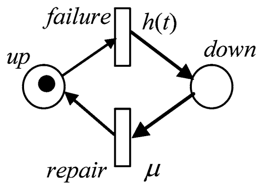


Fig. 10 Pure structure of the repairable model

Table 1 Algorithm of survivability analysis by SPN

Step 1:	*Modeling*. Establish service composition net model and service failure model for each service by SPN for Grid system;
Step 2:	*Composition*. Combine the same place of the above models, and build the composite model;
Step 3:	*Refinement*. Simplify the structure of an SPN model using transition enabling predicates and rate functions and exploit the near-independence of the SPN model;
Step 4:	*Solution*. Transform the SPN model to corresponding Markov process and solve the survivability attributes;
Step 5:	*Analysis*. Quantify the network survivability and analyze the subsistent problems.

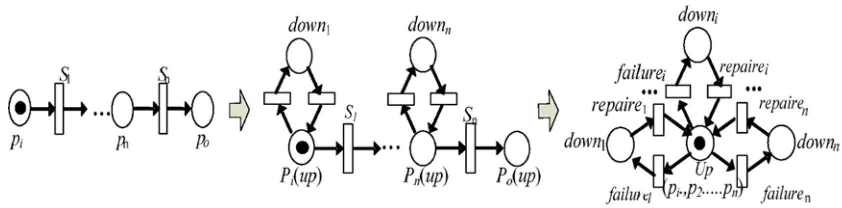


Fig. 11 The transition process of models

(or transition repair). So, the corresponding repairable model is given in Fig. 10. Generally, the time a transition firing satisfies the probability distribution, if we let the probability distribution be the exponential distribution, we can attain the intensity functions to the transitions implying the corresponding action. Therefore, in this model, we denote the intensity of the transition failure as $h(t)$ and the intensity of the transition repair as μ.

5.2 Survivability analysis framework

In this section, we will present a model and analysis framework for Grid service survivability based on the SPN. Table 1 shows the material algorithm.

Let us use the sequential service structure model to illustrate how to analyze the survivability. The process of modeling and transition can be shown as in Fig. 11. The examples of other model structures, such as parallel etc., can be found in literature [29].

After survivability SPN modeling, we can obtain the steady probability by the following processes.

First, according to the reachability graph, the homogeneous Markov chain (MC) corresponding to SPN model can be obtained. Second, define the state transfer matrix $Q = [q_{ij}]$, where q_{ij} $(i \neq j)$ is the transfer speed from state i to state j, if there is an arc between state i and state j. Third, we assume the steady probability of n states in MC is a row vector, $\Pi = (\pi_1, \pi_2, \ldots, \pi_n)$; according to the Markov process, we have the following linear simultaneous equations:

$$\begin{cases} \Pi \times Q = 0 \\ \sum_{i=0}^{n} \pi_i = 1 \end{cases}. \tag{19}$$

Finally, solving the above simultaneous equations, we can obtain the steady probabilities of the reachable marking, $P_i\ (t = \infty) = \pi_i$.

Based on the steady probability and the definitions of survivability attributes, we can quantify the survivability attributes.

5.3 Attribute quantification

In this section, we will introduce the quantification methods for some survivability attributes. According to the sequential structure example shown in Fig. 11, we can get reliability, MTTF, maintainability, MTTR, and availability.

Based on the methods mentioned above, we can get the MC corresponding to the model in Fig. 11 and the state transfer matrix Q:

$$Q = \begin{bmatrix} \Lambda & \lambda_1 & \ldots & \lambda_n \\ \mu_1 & -\mu_1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ \mu_n & 0 & \ldots & -\mu_n \end{bmatrix}$$

where $\Lambda = -\sum_{i=1}^{n} \lambda_i$. According to the linear simultaneous equations (19), we can get the steady probabilities when $t \to \infty$:

$$\begin{cases} \pi_0 = \left[1 + \sum_{i=1}^{n} \frac{\lambda_i}{\mu_i}\right]^{-1} \\ \pi_i = \pi_0 \frac{\lambda_i}{\mu_i}, \quad i = 1, \ldots, n \end{cases}. \tag{20}$$

(1) *Reliability*

When we compute the reliability, we do not need to compute all the solutions of $R(t) = P\{X > t\}$. Whether for the repairable or unrepairable system, we can always assume there are n states in the system. The former t consecutive normal work states are not absorbable states, and the last a states are absorbable states, they are fault states. Hence, we just think about the reliability of the system in absorbable states. Namely,

$$R(t) = P_0(t) = \exp(\Lambda t) = \exp\left(-\sum_{i=1}^{n} \lambda_i t\right). \tag{21}$$

(2) *MTTF*

When the failure rates of the services are exponentially distributed, $\lambda = \sum_{i=1}^{n} \lambda_i$, then

$$MTTF = \int_0^{\infty} R(t)dt = \int_0^{\infty} \exp\left(-\sum_{i=1}^{n} \lambda_i t\right) dt = \lambda^{-1}. \tag{22}$$

It can also be expressed by the average time in the state M_0.

(3) *Maintainability*

For maintainability, $M(t)$, we need to define the repairing rate $\mu(t)$ first. It describes the repairing probability within a unit time that the unrepairing services can be finished, when the repairing time has reached some time. The mathematical description is

$$\mu(t) = \frac{1}{1-M(t)}\frac{dM(t)}{dt}. \tag{23}$$

Taking integral of (23), we can get

$$M(t) = 1 - \exp\left(-\int_0^1 \mu(t)dt\right). \tag{24}$$

If $\mu(t) = \mu$ is a constant, then formula (24) becomes as follows:

$$M(t) = 1 - \exp(-\mu t). \tag{25}$$

(4) *MTTR*

MTTR denotes the average time to repair service nodes. It can be expressed as

$$MTTR = \lambda/\rho = \sum_{i=1}^{n} \lambda_i \Bigg/ \sum_{i=1}^{n} (\lambda_i/\mu_i) \tag{26}$$

where $\rho = \sum_{i=1}^{n} \rho_i$ is repairing coefficient of a whole system. $\rho_i = \lambda_i/\mu_i$ is repairing coefficient of the ith service.

(5) *Availability*

Based on the model in Fig. 11, when the system is in the state 0, that indicates that the service is available. So the steady availability can be obtained by the steady probability π_0. According to the computing result (23), A_s is as follows:

$$A_s = \pi_0 = \left(1 + \sum_{i=1}^{n} \frac{\lambda_i}{\mu_i}\right)^{-1} = \left(1 + \sum_{i=1}^{n} \rho_i\right)^{-1} = (1+\rho)^{-1}. \tag{27}$$

6 An example

To illustrate the service composition model and survivability analysis method, we use a travel example using Grid services that describe the way to travel abroad by means of some service composition. The detail descriptions about this application are as follows.

A user, named Aaron, plans to go on a journey to Europe during his vacation. For convenience, he chooses the travel Grid system to arrange all the matters concerned. Before the Grid system begins its service, Aaron needs to provide his requirements and the necessary information about himself. We assume he submitted his application at 10:10 am, on March 30th 2007. The whole travel case can be divided into four parts: submitting the application, selecting

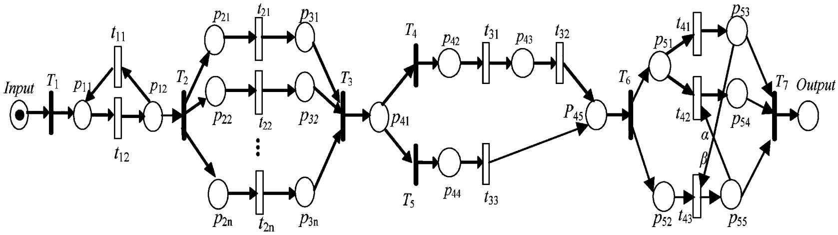


Fig. 12 *SCN* model of the travel Grid service example

the agent, getting the visa, and scheduling the flight and hotel. At the same time, he states his requirements, such as the start time and the end time, the total money he wants to pay, and so on. We assume he wants to pay \$100 for the travel Grid service. And he hopes he can get the result within 12 hours. So we get $TaskDe = (deadline = 18h;\ budget = \$100), SerDe = (SerID = 0603301010;$ $S = \langle applying, waiting, agent_{1,\ldots,n}, checking\ passport, checking\ visa, applying\ visa,$ $booking\ airplane_{1,2}, booking\ hotel\rangle)$. According to the description of the task, the corresponding service composition net can be shown as in Fig. 12.

Namely, $SCN = (P; T; F; \lambda, i; o)$, where $P = \{p_i, p_{11}, p_{12}, p_{21}, p_{22}, \ldots, p_{2n}, p_{31}, p_{32}, \ldots, p_{3n}, p_{41}, p_{42}, p_{43}, p_{44}, p_{45}, p_{51}, p_{52}, p_{53}, p_{54}, p_{55}, p_o\}$; $T = \{T_1, T_2, \ldots, T_7, t_{11}, t_{12}, t_{21}, t_{22}, \ldots, t_{2n}, t_{31}, t_{32}, t_{33}, t_{41}, t_{42}, t_{43}\}$; $F = \{(p_i, T_1), (T_1, p_{11}), (p_{11}, t_{12}), (t_{12}, p_{12}), (p_{12}, t_{11}), (t_{11}, p_{11}), (p_{12}, T_2), (T_2, p_{12}), (T_2, p_{22}), \ldots, (T_2, p_{2n}); (p_{21}, t_{21}), (t_{21}, p_{31}), (p_{22}, t_{22}), (t_{22}, p_{32}), \ldots, (p_{2n}, t_{2n}), (t_{2n}, p_{3n}), (p_{31}, T_3), (p_{32}, T_3), \ldots, (p_{3n}, T_3), (T_3, p_{41}), (p_{41}, T_4), (T_4, p_{42}), (p_{42}, t_{31}), (t_{31}, p_{43}), (p_{43}, t_{32}), (t_{32}, p_{45}), (p_{41}, T_5), (T_5, p_{44}), (p_{44}, t_{33}), (t_{33}, p_{45}), (p_{45}, T_6), (T_6, p_{51}), (T_6, p_{52}), (p_{51}, t_{41}), (p_{51}, t_{42}), (t_{41}, p_{53}), (t_{42}, p_{54}), (p_{52}, t_{43}), (t_{43}, p_{55}), (p_{53}, t_{43}), (p_{55}, t_{42}), (p_{53}, T_7), (p_{54}, T_7), (p_{55}, T_7), (T_7, p_o)$; $\lambda = \{\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \ldots, \lambda_{2n}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43}\}$. Here we assume the timed transitions are associated with exponential distributed firing times and the details are described as follows:

T_1 denotes the user begins to submit his request.
T_2 denotes the request is accepted, and the Grid begins to choose an appropriate agent for the user.
T_3 denotes the agent has been chosen, and the agent begins to check the condition of the user.
T_4 denotes the user has not gotten the visa or his visa is invalid, and needs to apply anew. We assume that from the statistic data, the transition T_4 can fire with probability $\alpha_1 = 0.7$.
T_5 denotes the user has the visa. We assume that from the statistic data, the transition T_5 can fire with probability $\alpha_2 = 0.3$.
T_6 denotes the visa is gotten.
T_7 denotes the flight and the hotel are scheduled successfully.
t_{12} denotes the user submits his request for Grid services, $\lambda_{12} = 10\, re/h$, it denotes that each service can complete 10 tasks per hour.
t_{11} is the course that the user waits for request submission, $\lambda_{11} = 12\, re/h$, we assume that from the statistic data, the transition t can fire with probability $\gamma = 0.3$.

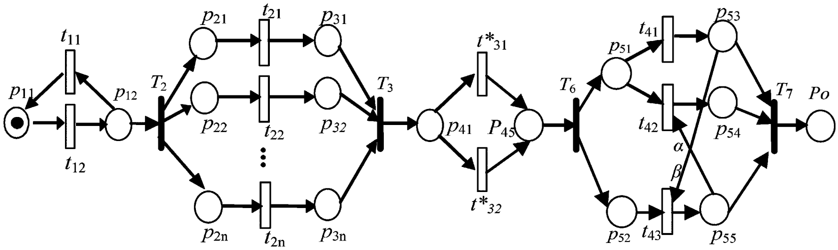
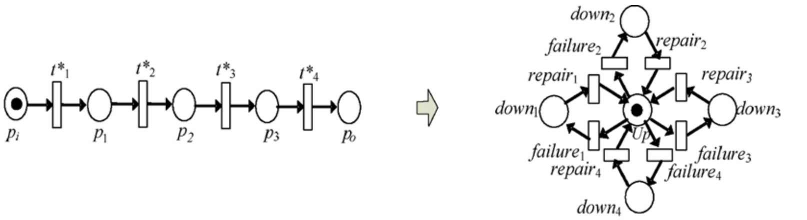


Fig. 13 Equivalent model of the Grid service example

$t_{21}, \ldots, t_{2n}$ denote the system looks for an appropriate agent for the user according to his requests. We assume $n = 3$ in this example, and $\lambda_{21} = 0.8\ re/h$, $\lambda_{22} = 0.7\ re/h$, $\lambda_{23} = 0.6\ re/h$.
t_{31} denotes checking the passport of the user, $\lambda_{31} = 0.2\ re/h$.
t_{32} denotes applying for the visa for the user, $\lambda_{32} = 0.05\ re/h$.
t_{33} models verifying the visa validity, $\lambda_{33} = 0.2\ re/h$.
t_{41} models buying the air ticket to the destination, $\lambda_{41} = 4\ re/h$.
t_{42} models booking the hotel, which sometimes must accord with the date of the air ticket, $\lambda_{42} = 4\ re/h$. We assume that from the statistical data, the firing of the transition t_{42} needs the firing of the transition t_{43} with probability $\alpha = 0.5$.
t_{43} models buying the return air ticket, which sometimes must accord to the free time date of the hotel, $\lambda_{43} = 5\ re/h$. We assume that from the statistical data, the firing of the transition t_{43} needs the firing of the transition t_{41} with probability $\beta = 0.5$.

If two places neighboring an immediate transition can be incorporated, it will not impact the computation of the models. That has been proved in the literature [25]. So we can leave out T_1, T_4, T_5 and incorporate the corresponding places. And then according to Theorem 1, we can get the equivalent transitions t^*_{31} and t^*_{32} from Fig. 11, and $\lambda^*_{31} = 0.04\ re/h$, $\lambda^*_{32} = 0.2\ re/h$. We can find the structures that are discussed above in Fig. 13. They are iterative service, parallel service, choice service and dependent service, respectively. At the same time, we can find if the user needs more agents in the second part, then the vote service will be necessary. Then we can simplify these services to the corresponding exterior services. The equivalent model can be shown as in Fig. 14(a). We get four equivalent exterior services and name the four compositional services as submission, agent, visa, schedule, namely $SerDe = (SerID = 0603301010;\ CS = \langle submission, agent, visa, schedule \rangle)$. Based on Theorems 2, 3, 4, 5, we can compute the equivalent service rates $\lambda^*_1 = 0.97\ re/h$, $\lambda^*_2 = 0.37\ re/h$, $\lambda^*_3 = 0.09\ re/h$, $\lambda^*_4 = 0.58\ re/h$. And similarly, according to Theorem 1, we can obtain the average service rate $\lambda = 0.15\ re/h$ and the whole response time of the Grid service for the travel schedule, $T = 16.56\ h$. If the average price of the service is $5\ D/h$, it will cost the user \$82.8. $ResDe = (NodeID, URL, P = 5\ D/h, T = 16.56\ h, C = \$82.8)$, where *NodeID* and *URL* should be two sets, they can be used to locate the resources and they only can be confirmed when the task is scheduled. Here we cannot make them certain.

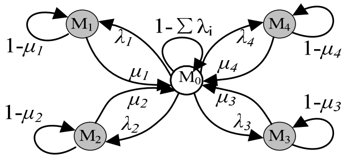
Using the methods mentioned above, we can compute the survivability attributes. Figure 14(b) denotes the corresponding service failure model of the equivalent sim-



(a) Equivalent simplification model (b) Corresponding service failure model

Fig. 14 The transition of the service failure model

Fig. 15 The corresponding Markov chain of the service failure model



plification model. And the corresponding Markov chain of the service failure model can be shown in Fig. 15.

Here $\lambda_{1,\ldots,4}$ are the failure firing rates and $\mu_{1,\ldots,4}$ are the repair firing rates. M_0 denotes the normal service states, $M_{1,\ldots,4}$ denote the service failure states. $X(t)$ is a Markov chain, the normal state is $\mathrm{E} = \{M_0\}$.

$$X(t) = \begin{cases} M_o & \text{the service in working order in time } t \\ M_i & \text{the service } i \text{ in fault, } i = 1, \ldots, 4, \text{ service failure in time } t \end{cases}.$$

According to statistical data, we can get some numerical values about these exterior services. If the failure service will be repaired immediately, when the fault takes place, then we can get the state transfer matrix Q

$$Q = \begin{bmatrix} \Lambda & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \mu_1 & -\mu_1 & 0 & 0 & 0 \\ \mu_2 & 0 & -\mu_2 & 0 & 0 \\ \mu_3 & 0 & 0 & -\mu_3 & 0 \\ \mu_4 & 0 & 0 & 0 & -\mu_4 \end{bmatrix}$$

where $\Lambda = -\sum_{i=1}^{n} \lambda_i$. According to the linear simultaneous equations (20), we can get the steady probabilities when $t \to \infty$:

$$\begin{cases} \pi_0 = \left[1 + \sum_{i=1}^{n} \frac{\lambda_i}{\mu_i}\right]^{-1} \\ \pi_i = \pi_0 \frac{\lambda_i}{\mu_i}, \quad i = 1, \ldots 4 \end{cases}.$$

Here we assume the failure and repair transitions are associated with exponential distributed firing time. And the failure firing rates $\lambda_1 = 0.1\ re/h$, $\lambda_2 = 0.02\ re/h$, $\lambda_3 = 2\ re/h$, $\lambda_4 = 0.8\ re/h$; the repair firing rates $\mu_1 = 0.5\ re/h$, $\mu_2 = 0.05\ re/h$, $\mu_3 = 2.1\ re/h$, $\mu_4 = 1\ re/h$. So $\pi_0 = 0.3, \pi_1 = 0.06, \pi_2 = 0.12, \pi_3 = 0.28, \pi_4 = 0.24$. Then we can compute some survivability attributes such as:

Reliability:

$$R(t) = \exp(\Lambda t) = \exp\left(-\sum_{i=1}^{4} \lambda_i t\right) = \mathrm{e}^{-2.92t},$$

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty \exp\left(-\sum_{i=1}^{4} \lambda_i t\right)dt = \left(\sum_{i=1}^{4} \lambda_i\right)^{-1} = 0.34\ h.$$

Maintainability:

$$M(t) = 1 - \exp\left(-\sum_{i=1}^{4} \mu_i t\right) = 1 - \mathrm{e}^{-3.65t},$$

$$MTTR = \frac{\sum_{i=1}^{4} \lambda_i}{\sum_{i=1}^{4} (\lambda_i/\mu_i)} = 1.24\ h.$$

Availability:

$$A_s = \pi_0 = \left(1 + \sum_{i=1}^{4} \frac{\lambda_i}{\mu_i}\right)^{-1} = 0.3.$$

The service failure model is based on different effects of all kinds of problems, such as attacks, faults and accidents. So it has embodied the effects of security incidents. If we want to describe the more detailed security behaviors, we need to construct a special security model, and the corresponding research work has been introduced in the literature [30].

In short, we can obtain the Grid interior service model for this travel task:

$GS_i = (TaskDe; SerDe; SCN; ResDe; Surv)$
$TaskDe = (deadline = 18\ h; budget = \$100)$
$SerDe = (SerID = 0603301010;\ S = \langle applying, waiting, agent_{1,\dots,n}, checking\ passport, checking\ visa, applying\ visa, booking\ airplane_{1,2}, booking\ hotel\rangle)$
$SerDe = (SerID = 0603301010; CS = \langle sub, agent, visa, sch\rangle)$
$SCN = (P; T; F; \lambda, i; o),$
$P = \{p_i, p_{11}, p_{12}, p_{21}, p_{22}, p_{23}, p_{31}, p_{32}, p_{33}, p_{41}, p_{42}, p_{43}, p_{44}, p_{45}, p_{51}, p_{52}, p_{53}, p_{54}, p_{55}, p_o\};$
$T = \{T_1, T_2, \dots, T_7, t_{11}, t_{12}, t_{21}, t_{22}, t_{23}, t_{31}, t_{32}, t_{33}, t_{41}, t_{42}, t_{43}\};$
$F = \{(p_i, T_1), (T_1, p_{11}), (p_{11}, t_{12}), (t_{12}, p_{12}), (p_{12}, t_{11}), (t_{11}, p_{11}), (p_{12}, T_2), (T_2, p_{12}), (T_2, p_{22}), (T_2, p_{23}); (p_{21}, t_{21}), (t_{21}, p_{31}), (p_{22}, t_{22}), (t_{22}, p_{32}), (p_{23}, t_{23}), (t_{23}, p_{33}), (p_{31}, T_3), (p_{32}, T_3), (p_{33}, T_3), (T_3, p_{41}), (p_{41}, T_4), (T_4, p_{42}), (p_{42}, t_{31}), (t_{31}, p_{43}), (p_{43}, t_{32}), (t_{32}, p_{45}), (p_{41}, T_5), (T_5, p_{44}), (p_{44}, t_{33}), (t_{33}, p_{45}), (p_{45}, T_6), (T_6, p_{51}), (T_6, p_{52}), (p_{51}, t_{41}), (p_{51}, t_{42}), (t_{41}, p_{53}), (t_{42}, p_{54}), (p_{52}, t_{43}), (t_{43}, p_{55}), (p_{53}, t_{43}), (p_{55}, t_{42}), (p_{53}, T_7), (p_{54}, T_7), (p_{55}, T_7), (T_7, p_o);$
$\lambda = \{\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43}\}.$
$ResDe = (NodeID, URL, P = 5\ D/h, T = 16.56\ h, C = \$82.8)$

$Surv = (R(t) = \mathrm{e}^{-2.92t},\ MTTF = 0.34\ h,\ M(t) = 1 - \mathrm{e}^{-3.65t},\ MTTR = 1.24\ h,$
$A_s = 0.3).$

The interior model has more details, and it is more suitable for the Grid service manager to manage and schedule the Grid resource. But it is too complicated for users. So we can show the corresponding exterior model to users, which can be shown as follows:

$GS_e = (TaskDe; SerDe; SCN; ResDe; Surv)$
$TaskDe = (deadline = 18\ h; budget = \$100)$
$SerDe = (SerID = 0603301010; CS = \langle submission, agent, visa, schedule\rangle)$
$SCN = (P; T; F; \lambda, i; o),$
$P = \{p_i, p_1, p_2, p_3, p_o\};$
$T = \{t_1^*, t_2^*, t_3^*, t_4^*\};$
$F = \{(p_i, t_1^*), (t_1^*, p_1), (p_1, t_2^*), (t_2^*, p_2), (p_2, t_3^*), (t_3^*, p_3), (p_3, t_4^*), (t_4^*, p_o);$
$\lambda = \{\lambda_1^*, \lambda_2^*, \lambda_3^*, \lambda_4^*\}.$
$ResDe = (NodeID, URL, P = 5\ D/h, T = 16.56\ h, C = \$82.8)$
$Surv = (R(t) = \mathrm{e}^{-2.92t},\ MTTF = 0.34\ h,\ M(t) = 1 - \mathrm{e}^{-3.65t},\ MTTR = 1.24\ h,$
$A_s = 0.3).$

Both users and the Grid service manager can find many useful data from the models. According to these data, users can know about the service state and ability; on the other hand, the Grid service manager can design different scheduling strategies, fault tolerance strategies and intrusion tolerance strategies. Consequently, the services can be guaranteed QoS for different user requests.

7 Conclusion and future work

In this paper, we have proposed a Grid service composition model and analysis method based on Stochastic Petri Nets. This method can fully capture the semantics of complex service compositions and their respective specificities. We divide the services into exterior service and interior service. The interior service contains all implementation information that the service manager needs, and it is not to be published to users. The exterior service can be obtained by simplifying the interior service. Some detailed information that users and service managers are concerned about are described. At the same time, we researched the equivalent simplification methods corresponding to these models and proved the simplification theorems, which are useful to minimize the state space of model and make models easy to compute. Though the results are approximately equivalent, the difference between the analysis result using this method and the accurate result using a traditional analysis technique is very small. Therefore, we conclude that this new method is concise and efficient. On the other hand, we investigate definitions of survivability and its main attributes, and quantifying method based on the model. It enhances the ability of the model to describe and measure the Grid service when attacks, faults and accidents happen. In the end, we use the method presented to describe and analyze an example of travel Grid services successfully.

In some complex and large-scale Grid systems, the whole model may not be simplified to a transition, which means these service composition models consist of more complicated patterns rather than the basic ones. However, there should be some submodels in the whole complex Grid system that could be analyzed using this method. Therefore, we can get a simplified equivalent model of the original one so that other traditional analysis techniques can be used in the simplified model. Our future work will focus on other typical complex Grid service composition patterns and integration of the model methods.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. 60803123, 60673187, 60673160 and 60673054) and China Postdoctoral Science Foundation funded project (No. 20080430040).

References

1. Mayer A, McGough S, Furmento N, Lee W, Gulamali M, Newhouse S, Darlington J (2004) Workflow expression: comparison of spatial and temporal approaches. In: Workflow in grid systems workshop, GGF-10, Berlin, March 2004
2. Foster I, Kesselman C, Nick MJ, Tuecke S (2002) The physiology of the grid: an open grid services architecture for distributed systems integration. Technical report, Argonne National Laboratory
3. Deelman E, Blythe J, Gil Y, Kesselman C (2003) Workflow management in GriPhyN. Grid resource management: state of the art and future trends, pp 99–116
4. Laszewski G, Zaluzec N, Hategan M, Rossi A (2003) GridAnt: Client side workflow management in grids with application onto position resolved diffraction. Midwest Software Engineering Conference, Chicago, US, June 2003
5. Alt M, Hoheisel A, Pohl HW, Gorlatch S (2006) A grid workflow language using high-level Petri nets. Lect Notes Comput Sci 3911:715–722
6. Neubauer F, Hoheisel A, Geiler J (2006) Workflow-based grid applications. Future Gener Comput Syst 22(1–2):6–15
7. Laprie JC, Kaaniche M, Kanoun K (1995) Modeling computer systems evolutions: non-stationary processes and stochastic Petri nets - application to dependability growth. In: Proceedings of the sixth international workshop on Petri nets and performance model (PNPM). October 1995, pp 221–230
8. Barry DK (2008) Web Services Flow Language (WSFL). http://www.service-architecture.com/web-services/articles/web_services_flow_language_wsfl.html
9. Jordan D, Evdemon J (2008) Web services business process execution language version 2.0. http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html, accessed on 10 November 2008
10. Krishnan S, Wagstrom P, von Laszewski G (2002) GSFL: a workflow framework for grid services. In: Proc. SC’2002, pp 11–16
11. van der Aalst WMP, Basten T (2002) Inheritance of workflows: an approach to tackling problems related to change. Theor Comput Sci 270(1–2):125–203
12. Hull R (2003) E-service composition: models and formalisms. In: Proceedings of the 2003 international workshop on description logics (DL2003), 81:1–14
13. Raman B, Agarwal S, Chen Y, Caesar M, Cui W (2002) The SAHARA model for service composition across multiple providers. Lect Notes Comput Sci 2414:585–597
14. Martin D, Paolucci M, McIlraith S, Burstein M, McDermott D, McGuinness D, Parsia B, Payne T, Sabou M, Solanki M, Srinivasan N, Sycara K (2005) Bringing semantics to web services: the OWL-S approach. In: 1st International workshop on semantic web services and web process composition (SWSWPC 2004). Lect Notes Comput Sci 3387:26–42
15. WSMO Working Group D16.1v0.2 (2005) The Web Service Modeling Language WSML, March 2005
16. Gronmo R, Jaeger MC (2005) Model-driven semantic web service composition. IEEE. Presented at the 12th Asia-Pacific software engineering conference (APSEC), Taipei, Taiwan. December 2005, pp 1–8

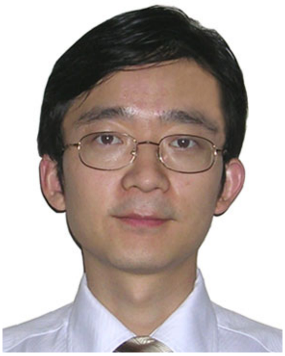
17. Berardi D, Calvanese D, Giacomo GD, Hull R, Mecella M (2005) Automatic composition of web services in Colombo. SEBD, pp 8–15
18. Zhovtobryukh D (2007) A Petri net-based approach for automated goal-driven web service composition. Simulation 83(1):33–63
19. Motahari-Nezhad HR, Li J, Stephenson B, Graupner S, Singhal S (2009) Solution marketplace for service composition and integration. In: 3rd International workshop on web service composition and adaptation, Los Angeles, US, July 2009
20. Hamadi R, Benatallah B (2003) A Petri net-based model for web service composition. The 14th Australasian Database conference, Adelaide, Australian, pp 191–200
21. Peterson JL (1981) Petri net theory and the modeling of systems. Englewood Cliffs, Prentice-Hall
22. Avizienis A, Laprie JC, Randell B, Landwehr C (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Secure Comput 1(1):11–33
23. Lin C, Qu Y, Ren FY, Marinescu DC (2002) Performance equivalent analysis of workflow systems based on stochastic Petri net models. International conference on engineering and deployment of cooperative information systems (EDCIS 2002). Lect Notes Comput Sci 2480:64–79
24. Ciardo G, Trivedi KS (1993) A decomposition approach for stochastic reward net models. Perform Eval 18(1):37–59
25. Lin C, Marinescu DC (1988) Stochastic high level Petri nets and applications. IEEE Trans Comput 37(7):815–825
26. Madan B, Goševa-Popstojanova K, Vaidyanathan K, Trivedi KS (2002) Modeling and quantification of security attributes of software systems. In: Proc. int'l conf. dependable systems and networks, pp 505–514
27. Zhang R, Heisig S, Moyle S, McKeever S (2005) Ogsa-based grid workload monitoring. In: Proceedings of the 5th IEEE/ACM international symposium on cluster computing and the grid (CCGRID'05). Cardiff, UK, May 2005, pp 668–675
28. Malhotra M, Trivedi KS (1995) Dependability modeling using Petri-nets. IEEE Trans Reliab 44(3):428–440
29. Lin C, Wang YZ, Yang Y, Qu Y (2006) Research on network dependability analysis methods based on stochastic Petri net. Acta Electronica Sinica 34(2):322–332 (in Chinese)
30. Lin C, Wang Y, Li QL (2005) Stochastic modeling and evaluation for network security. Chin J Comput 28(12):1943–1956

Yuanzhuo Wang (IEEE member) received his Ph.D. degree in Information Engineering School, University of Science and Technology, Beijing, China. He is currently a postdoctoral fellow at Computer Science Department, Tsinghua University, Beijing, China. His current research interests include service computing, network security analysis, performance evaluation, and stochastic game nets. He is a senior member of China Computer Federation. So far he has published over 50 papers.

Chuang Lin is a Professor and the Head of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the Ph.D. degree in Computer Science from Tsinghua University in 1994. In 1985–1986, he was a Visiting Scholar with the Department of Computer Sciences, Purdue University. In 1989–1990, he was a Visiting Research Fellow with the Department of Management Sciences and Information Systems, Graduate School of Business, University of Texas at Austin. In 1995–1996, he visited the Department of Computer Science, Hong Kong University of Science and Technology. His current research interests include computer networks, performance evaluation, network security analysis, logic reasoning, and Petri net theory and its applications. He has published more than 200 papers in research journals and IEEE conference proceedings in these areas and has published three books.

Professor Lin is a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP. He serves as the Technical Program Vice Chair, the 10th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004); the General Chair, ACM SIGCOMM Asia workshop 2005; the Associate Editor, IEEE Transactions on Vehicular Technology; and the Area Editor, Journal of Parallel and Distributed Computing.

Peter D. Ungsunan is a doctoral candidate in the Department of Computer Science and Technology at Tsinghua University. Previously he taught at the Beijing University of Aeronautics & Astronautics and was a network engineer at Lucent Technologies. He has M.Sc. and MBA degrees in CIS and Finance from the City University of New York and B.Sc. degree in Electrical and Mechanical Engineering from Rensselaer Polytechnic Institute.



Xiaomeng Huang received his Ph.D. degree of the Department of Computer Science and Technology, Tsinghua University, Beijing, China in 2007. He received a B.A. degree in 2000 from Wuhan University, Wuhan, China and a M.Sc. degree in 2003 from Huazhong University of Science and Technology, Wuhan, China. He is especially interested in distributed system, computer networks, etc.