

# Rationality of Service Composition of Workflow Net in a Service Oriented Architecture

Guangqi Huang, Li Huang, Xiuxiu Chen, Lily Sun

► **To cite this version:**

Guangqi Huang, Li Huang, Xiuxiu Chen, Lily Sun. Rationality of Service Composition of Workflow Net in a Service Oriented Architecture. Kecheng Liu; Stephen R. Gulliver; Weizi Li; Changrui Yu. 15th International Conference on Informatics and Semiotics in Organisations (ICISO), May 2014, Shanghai, China. Springer, IFIP Advances in Information and Communication Technology, AICT-426, pp.155-165, 2014, Service Science and Knowledge Innovation. <10.1007/978-3-642-55355-4\_16>. <hal-01350920>

**HAL Id: hal-01350920**

**<https://hal.inria.fr/hal-01350920>**

Submitted on 2 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Rationality of Service Composition of Workflow Net in a Service Oriented Architecture

Guangqi Huang<sup>1</sup>, Li Huang<sup>1</sup>, Xiuxiu Chen<sup>2</sup>, Lily Sun<sup>3</sup>

<sup>1</sup>Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan, China

<sup>2</sup>School of Management and Economics, Beijing Institute of Technology, Beijing, China

<sup>3</sup>School of Systems Engineering, University of Reading, UK  
{guangqi.huang, l.huang, Xiuxiu.Chen, lily.sun }@reading.ac.uk

**Abstract.** Users' requirements change drives an information system evolution. Consequently, such evolution affects those atomic services which provide functional operations from one state of their composition to another state of composition. A challenging issue associated with such evolution of the state of service composition is to ensure a resultant service composition remaining rational. This paper presents a method of Service Composition Atomic-Operation Set (SCAOS). SCAOS defines 2 classes of atomic operations and 13 kinds of basic service compositions to aid a state change process by using Workflow Net. The workflow net has algorithmic capabilities to compose the required services with rationality and maintain any changes to the services in a different composition also rational. This method can improve the adaptability to the ever changing business requirements of information systems in the dynamic environment.

**Keywords:** Service Composition, Service Composition Atomic-Operation Set, Rationality, Workflow Net, SOA

## 1 Introduction

With the extensive utilization of service oriented architecture (SOA) [1][2] in the field of information systems, the concept of service is employed in design and implementation of information systems. A software component can be described as technical service. A technical service consists of a composition of functional workflow which can be represented by an atomic-operation set. Technical services can be integrated to construct an information system. As [3] pointed out, technical services can be mapped on to business processes for meeting users' complicated requirements in various business domains [4].

In an information system application, its behavior and process of services composition evolve which is driven by orchestrating business services. Consequently, it affects the executable functional workflow to change from one stage of service composition to another state of composition. Therefore, the service composition process should have abilities to facilitate evolutions supported by the internal strategy and external environment [5-7]. The evolution of service composition refers to

services set or flow structure will add, update and adjust dynamically according to the changing requirements of business application during the implementation process. Generally, it includes two major forms, namely, services set change and service workflow structure adjustment [5][8]. In this paper, we focus on the methods to ensure the rationality of the process of service composition evolution behavior in SOA system.

Some achievements on the evolution of service composition have been made. Ref. [6] summarizes several common types of service composition evolution, including shallow and deep service evolution, and puts forward a service life cycle method. Ref. [9] proposes the solutions for dynamic composite service evolution problems from the research field of the trustworthy software, and discusses the existing work of dynamic service composition [10-11], including evolution time, evolution operation classification and evolution influence, etc. In ref.[12], the instance dynamic migration (IDM) of the Web service composition evolution is studied, the processing framework is built, and the optional rules and arithmetic to execute the IDM from the service choreography perspective are addressed. Regarding to the business process analysis and modelling, existing researches have illustrated series of WS-BPEL analysis methods and technologies based on Petri network [13-15], which provide the research method for this paper.

Therefore, this paper argues that how to correctly judge the rationality of business logic of the service composition behavior evolution is a basic problem in the SOA information systems. Based on the workflow-net theory, this paper studies the basic service composition operations according to the requirement in the SOA information system, puts forward the service composition atomic-operation set (SCAOS), and defines a kind of service composition atomic operations, which ensure the rationality of service composition process evolution. The following paper will be organized as follows. Section 2 illustrates the problems of the service composition. Section 3 describes the theory for analysis of the rationality of service composition. Section 4 proposes SCAOS while a sample is given to illustrate this process in section 5. Section 6 gives our conclusion and outlook.

## 2 Description of the Service Composition

Service composition problem is essentially a procedure that a set of services orchestrate according to the business process. It can be described as:  $C = \langle S : R \rangle$ ,  $C \in Service$ ,  $S \subset Service$ .  $C$  means the composite service,  $S$  means the set of atomic services,  $R$  means the relationship between the services in  $S$ . If the number of atomic services in  $S$  is 0 or 1,  $R$  will lose significance, service composition cannot be built. Therefore, in this paper, we define:

$$C \in CService \Leftrightarrow C = \langle \{se_0, se_1, \dots, se_n\} : R \rangle, se_i \in EService, 0 \leq i \leq n, n > 0$$

It means that the service  $C$  is a composite service, if and only if it is composed of two or more atomic services.

Generally, service composition behavior evolution of the SOA system is mainly caused by two factors: their services set and business process. Service composition behavior evolution caused by changes of services set can also be called  $S$  evolution of

service composition, denoted as  $E_s$ . In this process, the atomic services set  $S$  changes, which results in a change of  $C$ . The evolution caused by changes of business process is also called as  $R$  evolution of service composition, denoted as  $E_R$ . In this process, the atomic services set  $S$  is not changed, while the relationship  $R$  changes, which results in a change of  $C$ .

### 3 Theory for Analysis on the Rationality of Service Composition Operation

The use of workflow net (WF-net) to describe the service composition and analysis the rationality of service composition behavior evolution can be defined as follows:

**Definition 1.** A Petri net  $WFN = (P, T, F)$  is called as WF-net [9][16], if and only if:

- Exists an initial place  $i \in P$ , its precursor is empty,  $\bullet i = \emptyset$ .
- Exists an end place  $o \in P$ , its successor is empty,  $o \bullet = \emptyset$ .
- For any node  $x \in P \cup T$ , all belongs to a path from  $i$  to  $o$ .

WF-net can be used to describe the internal logic of service composition dynamically.

For  $WFN = (P, T, F)$ ,

- The set of place  $P$  is used to describe the conditions of service calling.
- The set of transition  $T$  is used to describe the collection of service units of composite service, it is actually an operation.
- The set of flow relationship  $F$  is used to describe the logical relationships between service units of composite service.

The distribution of the Tokens in all places is identified as network state, initial state (final state) of WF-net denoted as  $M_0(M_{end})$ . In  $M_0(M_{end})$ , only the initial place (the end place) is marked.

**Definition 2. Rational WF-net [9].** A WF-net  $WFN = (P, T, F, i, o, M_0)$  is rational, if and only if:

(1) For any state  $M$  can be reached from the initial state  $M_0$ . There is a transition sequence that drives the state  $M$  to the end state  $M_{end}$ , the formalization description is:

$$\forall M (M_0 \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} M_{end})$$

(2) The end state  $M_{end}$  is the only final state which can be reached from the initial state  $M_0$ , and  $M_{end}$  will have at least one mark, the formalization description is:

$$\forall M (M_0 \xrightarrow{*} M \wedge M \geq M_{end}) \Rightarrow (M = M_{end})$$

(3) Dead transition does not exist in the WF-net, the formalization description is:

$$\forall t \in T, \exists M, M' \text{ s.t. } M_0 \xrightarrow{*} M \xrightarrow{t} M'$$

The rationality of a WF-net, can be used to judge whether a service composition behavior evolution is rational:

- The rationality of a WF-net can ensure that the evolved process of service composition can be smoothly completed.
- Internal constitution service should all stop until the end of service composition.
- All internal constitution service of service composition could be called.

**Theorem 1.** For WF-net  $WFN_1 = (P_1, T_1, F_1)$  and  $WFN_2 = (P_2, T_2, F_2)$ ,  $T_1 \cap T_2 = \emptyset$ ,  $P_1 \cap P_2 = \{i, o\}$  and  $t^* \in F_1$ . The WF-net  $WFN_3 = (P_3, T_3, F_3)$  which replaces the transition  $t^*$  of  $WFN_1$  with  $WFN_2$  can be built with  $F_3 = \{(x, y) \in F_1 \mid x \neq t^* \wedge y \neq t^*\} \cup \{(x, y) \in F_2 \mid (x, y) \cap \{i, o\} = \emptyset\} \cup \{(x, y) \in P_1 \times T_2 \mid (x, t^*) \in F_1 \wedge (i, y) \in F_2\} \cup \{(x, y) \in T_2 \times P_1 \mid (t^*, y) \in F_1 \wedge (x, o) \in F_2\}$ ,  $P_3 = P_1 \cup P_2$ ,  $T_3 = (T_1 \setminus \{t^*\}) \cup T_2$ . Therefore,  $WFN_1$  and  $WFN_2$  are rational if and only if  $WFN_3$  is rational. Proof of this theorem is proved by [7].

## 4 Service Composition Atomic-Operation Set

Based on the WF-net theory of service composition description, this paper puts forward the service composition atomic-operation set (SCAOS). Faced with business logic evolution requirements in the information system, this paper defines 2 classes and 13 kinds of basic service composition of atomic operations. It can be used to specify the service composition behavior in SOA system, in order to guarantee the rationality of evolution process of service composition behavior. The service composition that can be described by rational WF-net is also called as rational service composition.

### 4.1 Internal Relationships in the Service Composition

Without considering the cycle of calls, the relationships between service units of composite service include the following three types: sequence, switch and flow. Here, we define that  $\forall a, b \in S$  are represented by the transition  $t_a$  and transition  $t_b$ , respectively. Therefore,

- (1) Sequence: Two service units are sequentially called, that is

$\exists p_s \in P$  s.t.  $p_s \in t_a \wedge p_s \in t_b$ , as shown in Fig. 1:

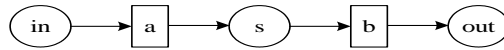


Fig. 1. Sequence relationship

- (2) Switch: Two service units are selected to call according to the specific condition, that is  $\exists p_{in}, p_{out} \in P$  s.t.  $p_{in} \in t_a \wedge p_{in} \in t_b \wedge p_{out} \in t_a \wedge p_{out} \in t_b$  (shown in Fig. 2)

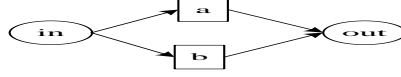


Fig. 2. Switch relationship

- (3) Flow: Two service units are called at the same time (shown in Fig. 3), that is

$$\exists p_{in}, p_{in'}, p_{out'}, p_{out} \in P \wedge t_{split}, t_{join} \in T, s.t.$$

$$\{p_{in}, p_{in'}\} \subset \dot{t}_{split} \wedge p_{in} \in \dot{t}_a \wedge p_{in'} \in \dot{t}_b \wedge p_{out'} \in \dot{t}_a \wedge p_{out} \in \dot{t}_b \wedge \{p_{out}, p_{out'}\} \subset \dot{t}_{join}.$$

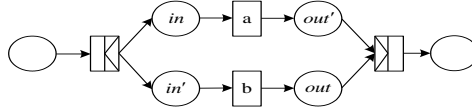


Fig. 3. Flow relationship

#### 4.2 Evolution of Service Composition Behavior Caused by the Change of Services Set

For composite service  $C = \langle S : R \rangle$ , service composition behavior evolution caused by the change of services set can be defined as follows:

**Definition 3.** *S* Evolution  $E_S$ : Refers to the composite services set  $S = dom(C)$  changes, which also led to a change in the relationship. It includes two aspects, namely, increase and decrease. Considering the above three kinds of basic internal relationships of service composition, it is divided into the following 6 kinds of evolution:

- (1) Sequence Increase Evolution  $b_{sq^+}$  (Shown in Fig. 4):

$$T^N = T^O \cup \{t_b\} \wedge P^N = P^O \cup \{p_s\} \wedge F^N = F^O \cup \{(t_a, p_s), (p_s, t_b), (t_b, p_{out})\} \setminus \{(t_a, p_{out})\}$$

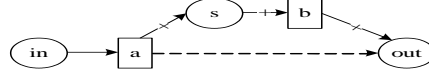


Fig. 4. Sequence Increase Evolution

- (2) Switch Increase Evolution  $b_{sw^+}$  (Shown in Fig. 5):

$$T^N = T^O \cup \{t_b\} \wedge P^N = P^O \wedge F^N = F^O \cup \{(t_b, p_{out}), (p_{in}, t_b)\}$$

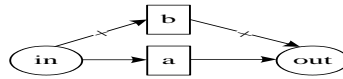


Fig. 5. Switch Increase Evolution

- (3) Flow Increase Evolution  $b_{fl^+}$  (Shown in Fig. 6):

$$\begin{aligned} T^N &= T^O \cup \{t_b\} \wedge P^N = P^O \cup \{p_{in'}, p_{out'}\} \wedge F^N \\ &= F^O \cup \{(t_{split}, p_{in'}), (p_{in'}, t_b), (t_b, p_{out}), (t_a, p_{out'}), (p_{out'}, t_{join})\} \setminus \{(t_a, p_{out})\} \end{aligned}$$

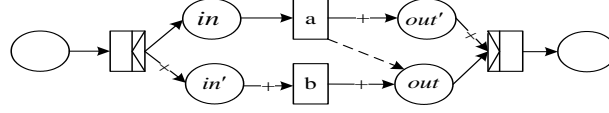


Fig. 6. Flow increase Evolution

The inverse evolution behavior of the above evolutions can be defined as: Sequence Decrease Evolution  $b_{sq^-}$ , Switch Decrease  $b_{sw^-}$ , and Flow Decrease Evolution  $b_{fl^-}$ . They are similar with the above evolutions, so here we do not describe them in detail.

### 4.3 Service composition behavior evolution caused by the business process change

For composite service  $C = \langle S : R \rangle$ , service composition behavior evolution caused by changes of business process can be defined as follows:

**Definition 4.**  $R$  Evolution ( $E_R$ ): Refers to the relationship  $R$  changes but the composite services set  $S = dom(C)$  remains. In other words, the service business process is changed.

According to the actual requirement of service composition behavior evolution, considering the above three kinds of basic internal relationships of service composition, 7 kinds of evolutions can be got as follows, as shown in Fig. 7:

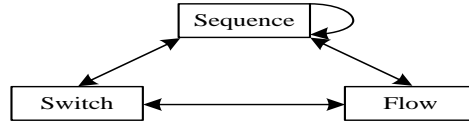


Fig. 7. 7 kinds of  $R$  evolution behaviors (indicated by the directions of arrows)

Define the relative start and end WF-net of evolution as  $WFN^O = (P^O, T^O, F^O, i^O, o^O, M_0^O)$  and  $WFN^N = (P^N, T^N, F^N, i^N, o^N, M_0^N)$ ,  $WFN^O$  is a rational WF-net,  $A \setminus B$  means set out the elements like set B from set A, then the following 7 kind of evolution behaviors maybe occur in  $WFN^O \rightarrow WFN^N$ :

- (1) Reverse Evolution  $b_{sq2sq}$ : the sequence execution of service  $a$  and service  $b$  is reversed, described as:  

$$T^N = T^O \wedge P^N = P^O \wedge F^N = F^O \cup \{(t_b, p_s), (p_s, t_a)\} \setminus \{(t_a, p_s), (p_s, t_b)\}$$
- (2) Sequence to Switch Evolution  $b_{sq2sw}$ : the sequence execution of service  $a$  and service  $b$  is changed to switch execution, described as:  

$$T^N = T \wedge P^N = P^O \setminus \{p_s\} \wedge F^N = F^O \cup \{(p_{in}, t_b), (t_a, p_{out})\} \setminus \{(t_a, p_s), (p_s, t_b)\}$$
- (3) Sequence to Flow Evolution  $b_{sq2fl}$ : the sequence execution of service  $a$  and service  $b$  is changed to flow execution, described as:

$$\begin{aligned}
T^N &= T^O \wedge P^N = P^O \cup \{p_{in'}, p_{out'}\} \setminus \{p_s\} \wedge F^N \\
&= F^O \cup \{(t_{split}, p_{in'}), (t_a, p_{out'}), (p_{in'}, t_b), (p_{out'}, t_{join})\} \setminus \{(t_a, p_s), (p_s, t_b)\}
\end{aligned}$$

(4) Switch to Flow Evolution  $b_{sw2fl}$ : the switch execution of service  $a$  and service  $b$  is changed to flow execution, here  $\{t_{split}, t_{join}\} \subset T^O$ , described as:

$$\begin{aligned}
T^N &= T^O \wedge P^N = P^O \cup \{p_{in'}, p_{out'}\} \wedge F^N \\
&= F^O \cup \{(t_{split}, p_{in'}), (p_{in'}, t_b), (t_a, p_{out'}), (p_{out'}, t_{join})\} \setminus \{(p_{in'}, t_b), (t_a, p_{out'})\}
\end{aligned}$$

The inverse evolution behavior of the last three evolutions mentioned above can be defined as: Switch to Sequence Evolution  $b_{sw2sq}$ , Flow to Sequence Evolution  $b_{fl2sq}$ , and Flow to Switch Evolution  $b_{fl2sw}$ . They are similar with the above evolutions, so we do not describe them here in detail.

#### 4.4 Rationality analysis of service composition behavior evolution

The verification of the rationality of operations of service composition behavior evolution can be transferred to the verification of rationality of relative WF-net. It is proved that the service composition atomic-operation set can keep the rationality of evolved composite service. The following propositions are true:

**Proposition 1.** *Atomic-operation set of S Evolution,  $E_S = \{b_{sq^+}, b_{sq^-}, b_{sw^+}, b_{sw^-}, b_{fl^+}, b_{fl^-}\}$  can keep the rationality of evolved composite service.*

**Proposition 2.** *Atomic-operation set of R Evolution,  $E_R = \{b_{sq2sq}, b_{sq2sw}, b_{sw2sq}, b_{sq2fl}, b_{fl2sq}, b_{sw2fl}, b_{fl2sw}\}$  can keep the rationality of evolved composite service.*

In this paper, we take the switch to flow evolution  $b_{sw2fl}$  as an example to prove the above propositions.

*Proof:* According to the switch relationship, a WF-net  $WFN_{sw}^O$  is built,  $P_{sw}^O = \{i, o, p_{in}, p_{out}\}$ ,  $T_{sw}^O = \{t_{split}, t_{join}, t_a, t_b\}$ ,  $F_{sw}^O = \{(i, t_{split}), (t_{split}, p_{in}), \dots, (t_{join}, o)\}$  as shown in Fig. 8a. Similarly, according to the flow relationship, the other one  $WFN_{fl}^N$  is built, as shown in Fig. 8b.

According to Theorem 1, the WF-net before evolution  $WFN^O = WFN_{sw}^O + WFN_{other}$  can be got.  $WFN_{other}$  is the residual network of the WF-net before evolution, is similar with the  $WFN_1$  in [Theorem 1].  $WFN^O$  is rational, so  $WFN_{other}$  is rational. Obviously that the WF-net  $WFN_{fl}^N$  is rational, then the evolved WF-net  $WFN^N = WFN_{fl}^N + WFN_{other}$  is rational too.



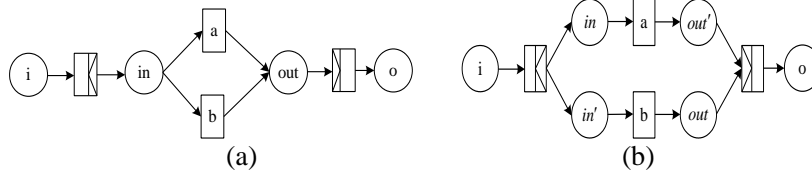


Fig. 8. Proof of the rationality of evolution behavior

#### 4.5 Rationality analysis of mixed service composition behavior evolution operations

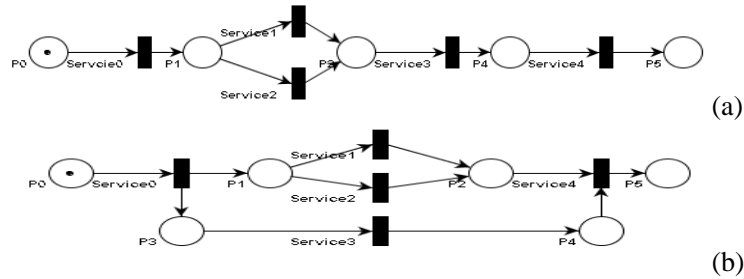
In the application of actual SOA system, a whole service composition behavior can be decomposed into multiple mixed service composition atomic operations, so a service composition behavior is essentially an operation sequence of atomic operations. Based on [Proposition 1] and [Proposition 2], it is guaranteed that the rationality of the whole service composition behavior remains unchanged. In addition, for the nested service composition behavior, according to [Theorem 1], if we transfer the transitions  $t_a$  and  $t_b$  of S evolution and R evolution behavior and replaced it with a rational WF-net, then the evolved WF-net is still rational. It is consistent with the inclusion of service, so a rational WF-net corresponds to a rational composite service. This rational composite service can be used as a basic service unit to participate higher level service composition, and this evolution behavior is rational too.

In summary, for the defined service composition atomic-operation set in the SOA system  $SCAOS = E_S \cup E_R$ , every operation  $\forall b \in SCAOS$ , the evolved service composition process is rational.

### 5 Sample of Service Composition Behavior Evolution

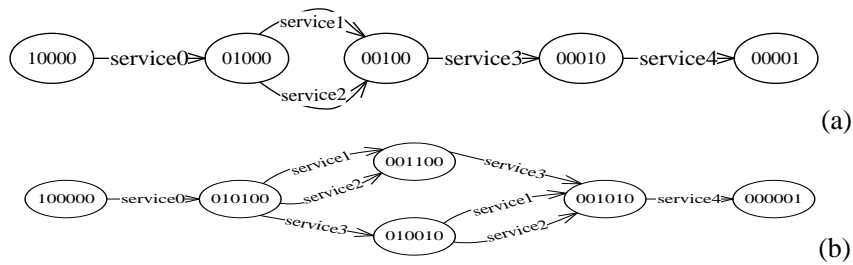
Suppose in the information service center of a bank system, the user information query process is implemented as a composite service. Its logic process is designed as follows: firstly, input the user account code, and call user identification service **Service0**; then, according to the category of user account, call the VIP user account financial information query service **Service1** or common user account financial information query service **Service2**; subsequently, call user basic information query service **Service3**; and finally, call user information comprehensive analysis and processing service **Service4**, return the results of user account information analysis.

We found that if the bank system query the account financial information using **Service1** or **Service2** and query the basic information using **Service3** at the same time, the reaction rate of user account information analysis will maybe increase. Thus the original business logic is changed and the service composition process evolution happens. The relationships among **Service3** and selective **Service1** and **Service2** are the order of sequence execution before evolution, while their relationships are changed to the order of concurrent execution after evolution, as shown in Fig. 9:



**Fig. 9.** Relative WF-net before evolution (a) and after evolution (b)

Thus, the above service composition evolution is essentially a kind of R evolution - Sequence to Flow Evolution  $b_{sq2fl}$ . It is rational. Then, we build the state reachability graph of the relative WF-net before evolution and after evolution, as shown in Fig. 10:



**Fig. 10.** State reachability graph before evolution (a) and after evolution (b)

Comparing the state reachability graph before evolution and after evolution, obviously the evolution behavior is rational and the results of the business processes are consistent.

## 6 Conclusions

A challenging issue associated with service composition is to remain the rationality of service composition. In order to ensure the rationality of the service composition behavior, this paper studies the service composition operations according to the requirement in the SOA information system, puts forward the SCAOS, defines 2 classes and 13 kinds of basic service composition atomic operations, including service composition operations caused by the changes of services set and business process. Specifically, S evolution operations includes Sequence Increase Evolution, Switch Increase Evolution, Flow Increase Evolution, Sequence Decrease Evolution, Switch Decrease and Flow Decrease Evolution; R evolution operations includes Reverse Evolution, Sequence to Switch Evolution, Sequence to Flow Evolution, Switch to Flow Evolution, Switch to Sequence Evolution, Flow to Sequence Evolution and Flow to Switch Evolution.

Using the WF-net theory, this paper proves that SCAOS can maintain the rationality of service composition operations of SOA system when the service composition behavior occurs. The workflow net has algorithmic capabilities to compose the required services with rationality and maintain that any changes to the services in different compositions are also rational. SCAOS remains the rationality of the internal process logic in the SOA information system. This method can enhance the adaptability of SOA information systems to respond to the ever changing business requirements in the dynamic environment.

## References

1. Eric Newcomer, Greg Lomow: Understanding SOA with Web Services, Addison Wesley Professional (2004).
2. Chai Xiaolu, Liang Yuqi: Web Services technology, architecture and Application. Electronics Industry Press, Beijing (2002).
3. Yu Jian, Han Yanbo. Service oriented computing-principle and application. Tsinghua university press, Beijing (2006).
4. Zhang Zhongyu, Huang Guangqi: Transparency of Information System Service Evolution, Computer Engineering and Design, 32(2011).
5. Ryu S H, Casati F, Skogsrud H, Benatallah B, Saint-Paul R: Supporting the dynamic evolution of Web service protocols in service-oriented architectures. ACM Transactions on the Web, 2(2008) 1-45.
6. Papazoglou M P.: The challenges of service evolution. International Conference of Advanced Information Systems Engineering, (2008) 1-15.
7. Zhang Zhongyu, Huang Guangqi, Chang Tao, Zhang Pei: Approach of QoS-Oriented Services Run-Time Binding, Computer Engineering and Design, 32(2011) 2700-2703.
8. Huang Guangqi, Cai Chuanqing, Shen Chenping, Zhang Zhongyu: Study on Evolution Behavior of Service Combination under Unchanged Set, Compute Science, 39 (2012).
9. Zeng Jin, Sun Hailong, Liu Xudong, Deng Ting, Huai Jinpeng: Dynamic evolution mechanism for trustworthy software based on service composition, Institute of Software, 21(2010) 261-276.
10. Von de Aalst WMP, Jablonski S: Dealing with workflow change: Identification of issues and solutions. Int'l Journal of Computer System Science & Engineering, 15 (2000) 267-276.
11. Andrikopoulos V, Benbernou S, Papazoglou M.P.: Managing the evolution of service specifications. The 20th Int'l Conf. on Advanced Information Systems Engineering, (2008) 359-374.
12. Song Wei, Ma Xiaoxing, Lu Jian, Instance migration in dynamic evolution of web service compositions, Chinese Journals of Computers, 32(2009) 1816-1831.
13. Niels Lohmann.: A Feature-Complete Petri Net Semantics for WS-BPEL 2.0, Lecture Notes in Computer Science, (2008) 77-91.
14. Niels Lohmann, Wolf. K.: How to implement a theory of correctness in the area of business processes and services. 8th International Conference on Business Process Management, (2010) 61-77.
15. Liske N, Niels Lohmann, Stahl C, Wolf K. Another Approach to Service Instance, Service-Oriented Computing. 7th International Joint Conference, (2009) 24-27.
16. Van der Aalst W MP. Workflow Management Models, Methods, and Systems, the MIT Press Cambridge, Massachusetts London, (2002).