

The Behavior Trustworthiness Analysis Methods of Composite Web Services Based on Process Mining

Xiangwei Liu¹, Xianwen Fang^{2,*}, Mimi Wang² and Zhixiang Yin²

¹ Department of Information Management, Anhui University of Science and Technology, Huainan 232001, China

² Department of Information and Computing Science, Anhui University of Science and Technology, Huainan 232001, China

Received: 31 Jan. 2013, Revised: 1 Jun. 2013, Accepted: 2 Jun. 2013

Published online: 1 Sep. 2013

Abstract: At present, the behavior trustworthiness of the composite Web service was very important. The soundness of the composite Web service model was the core of the behavior trustworthiness of the service composition, which includes two steps, one was to determine whether or not the composite model satisfied the behavior relativity, in order to analyze the function predictability; the other was to compare the behavior conformance for analyzing the influence of external factors. But the proposed judging algorithm of behavior relativity was feasible only in the case that T-invariant of Petri net model exists, and the behavior conformance was also difficult to handle. In the paper, firstly, in order to analyze the behavior relativity between the Web services, an algorithm for determining the behavior weak soundness of the composite service was presented based on the theory of the service tree. Then, we present an effective analysis method of behavior conformance based on process mining. Finally, we develop plug-ins and test our methods in large benchmarks, and compare our methods with genetic process mining methods. The theoretical analysis and experimental results indicate that this method can realize the predictable behavior of software well, and satisfy the behavior trustworthiness requirements of software.

Keywords: Behavior Trustworthiness; Process Mining; Open Petri Net; Web service; Behavior Relativity.

1 Introduction

With the tremendous growth of information and communication technologies, it is advantageous to design and implement the complex inter-enterprise Web service processes. One of the major innovations is the concept of service-oriented architectures (SOA) which considers software systems as being made up with autonomous, dynamic, loosely coupled and service-based components. As a new computational resource configuration in the Internet, Web services played a more and more important role in the field of the electronic commerce, enterprise application integration and so on. Many researchers pointed out that services do not only contain static syntax and semantics, but also contain some dynamic behavior attributes, e.g. internal control flow, data flow and interaction protocol. Now the service composition approach becomes mature, so a large number of large-granularity process web service compositions will occur. These services implied business process a logic and the operations offered have some temporal relation.

For interacting Web service components, especially those that run over the Internet, behavior trustworthiness plays a crucial role. The operating and development environment of Web service components turn from the traditional closed static environment and extend to an open, dynamic, ever-changing network environment[1]. In the network computing environment, the behavior of the entity is uncontrollable and uncertain, which posed a severe challenge for behavior trustworthiness of the running processes[2]. Therefore, the guarantee of the behavior trustworthiness of the composite Web service under the network environment has become the industry's research focus. Hence, it is key to ensure the behavior trustworthiness after compositions except ensuring the correct function of every service.

Analysts often see lack of trustworthy methods as a major impediment to the adoption of web services in building agile business process. The behavior of Web service is a running operation series, which can describe some feature when one Web service component interacting with others. By analyzing the definition of

* Corresponding author e-mail: fangxianwen@hotmail.com

trustworthiness, that the behavior can be predictable is a key property. So, achieving behavior trustworthy goals often relies on an appropriate use of Web service behavior. However, the notion of behavior trustworthiness is often neglected in composite Web services, which usually concentrates on modeling the process in a way that functional correctness and security can be shown, either manually or using formal proof tools like model checking and some static validation methods. In contrast to features that are crucial for functional correctness and security features, and so on, behavior trustworthiness is typically integrated in an application, which mainly assures the execution behavior and computing result of the Web service to be consistent with the expected effects. The integration of trustworthy features into a composite Web service is not well understood. Especially, the trustworthy features are difficult to describe, and to analyze quantitatively[3]. So, the problem need further investigation, and seek new methods to deal with the behavior trustworthiness.

The composite Web service may run into two kinds of problems, one problem is the interactive influence between the Web service components in the composition process; one Web service component may influence the others, which leads to the function of composite Web service to be unpredictable. The other problem is that the composite Web service process may be influenced by the external factors. Aimed to the internal interactive influence, the interaction behavior relativity of the composite Web service need be studied. At present, the correctness of Web service composition was mainly determined by the soundness and the behavior soundness attracts many researchers attentions. Most of these researches were based on Petri Net, finite state machine or automata theory. Based on the concept of service view, Fang et al proposed a formal definition of behavior relativity between services, and then presented a Petri net method to qualitatively determine and quantitatively compute behavior compatibility. Foster[4] translated the service composition into the finite state machine (FSM), and used labeled transition system analyzer to analyze FSM model, then analyzed behavior compatibility among services by verifying its safety, live and deadlock property. Fu[5] modeled service composition Business Process Execution Language (BPEL) description by Guarded automata, used temporal logic to describe the goal attribute which is required to meet the interaction process. Then, translated the model into the Promela specification and used Simple Promela Interpreter (SPIN) model checker to verify whether the model meets goal attribute property and it is a high demand for determining the correctness of Web service composition. Martens[6] applied the workflow net to model Web services and analyze the service behavior compatibility in the composition by verifying safety, live and deadlock property in the Workflow Net. Based on extended the Martin-Lf's type theory (MTT) which supports a type-theoretic formulation of services behavior structured

patterns, Yin[7] proposed the verification on consistency and compatibility of Web services behavior. Fang[8] studied the soundness property from the language viewpoint of the workflow net system. Net language depiction for soundness was presented to reveal the behavior characteristic of this basic property. Furthermore, synchronous composition and combined composition of the workflow net system was studied. Necessary and sufficient conditions were given for soundness preservation whether the two composition system. This study provided a novel approach to the modeling and analysis for composition in complex WF-net systems. However, it was difficult to determine the soundness property and it was a high demand for determining the correctness of Web service composition. In the paper, though discovering the limitation of the method based on behavior relativity to judge the soundness, a decision method of behavior weak soundness is proposed to determine the correctness of composite Web service.

For the external factors, we use the Web service process mining methods to build the actual behavior model based on the running logs. At present, some process mining methods[3, 9, 10, 11] have been proposed, one is the abstraction-based mining algorithms, the algorithms construct a net based on an abstraction of the log and most of them are derived from the α -algorithm[3, 9], because the mining procedures of these algorithms are very similar, called as α -series algorithms. But, α -series algorithms only can mine the WF-net, which is a safe net, and there is still not a single abstraction-based algorithm that can handle all the special constructs in a sound SWF-net. The theory of regions can be used to transform a state-based model or a set of words into a Petri net that exactly mimics the behavior given as input. Recently several papers appeared on the application of the theory of regions for process discovery. Two types of region theory can be distinguished, namely state-based region theory and language-based region theory[11]. The region-based process mining algorithms can mine the no-safe net, the algorithms obtain the places based on the minimal regions, then construct the Petri net model by analyzing the transitions. But the constructing process only considers the direct causal dependency, some indirect causal dependency relationships are not taken into account. The genetic process mining algorithms[12] can deal with noise and can be used to express the main behavior (i.e., not all the details and exceptions) registered in an event log. It supports the mining of all common constructs in process models (i.e., sequence, choice, parallelism, loops, invisible tasks and some kinds of non-free-choice), except for duplicate tasks. But the mining result is main behavior model, so some low occur frequency behavior may be not mined. But the existing methods are deficient to solve the problem. In the paper, our contributions are given out the analyzing methods of the behavior trustworthiness about composite Web service based on the induction information; the methods can deal

with k-bounded Web service process, and take the indirect dependent relationships into account.

2 The analysis method about behavior weak soundness of web services based on Petri Net

A Web service consists of internal structures that realize a local sub process and an interface to communicate with its environment. So modeling services with the help of Open Petri Net (OPN), i.e. a Petri Net with an interface. Then, giving out the definition of OPN.

Definition 1[13](Petri Net) A 3-tuple $N = (P, T; F)$ is called Petri Net, where

- (1) $P \cup T \neq \emptyset$
- (2) $p \cap T \neq \emptyset$
- (3) $F \subseteq ((F \times T) \cup (T \times P))$
- (4) $\text{dom}(F) \cup \text{cod}(F) = P \cup T$
 $\text{dom}(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (x, y) \in F\}$
 $\text{cod}(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (x, y) \in F\}$

Definition 2[13](Open Petri Net) An Open Petri Net is a 7-tuple $N = (P, I, O, T, F, i, f)$, where

- (1) $(P \cup I \cup O, T, F)$ is a Petri net;
- (2) P is a set of internal places; T is a set of transitions; F is a set of flow relations;
- (3) I is a set of input places, and $\bullet I = \emptyset$; O is a set of output places, and $\bullet O = \emptyset$;
- (4) i is the initial marking;
- (5) f is the final marking, and f is a deadlock.

We call the set $I \cup O$ the interface places of the OPN. Note that the initial and final markings cannot mark interface places.

Definition 3(Weak Sound) An OPN N is called weak sound, if and only if:

- (1) For any marking $\forall m \in R(N, i_N)$, holds $S(N) : m \xrightarrow{*} f_N$, i.e. the final marking f_N is reachable.
- (2) For any marking $\forall m \in R(N, i_N)$, such that $m \geq f_N$ holds $m = f_N$.

Two OPNs can be composed fusing interface places with the same name. Next, we give the condition of two OPNs composed and their composition.

Definition 4[14] (Composable) Two OPNs A and B are composable if and only if they do not share any internal places, input places and output places, i.e.

$$(P_A \cup P_A \cup O_A \cup T_A) \cap (P_B \cup P_B \cup O_B \cup T_B) = (I_A \cap O_B) \cup (O_A \cap I_B) \text{ hold.}$$

Definition 5[14] (Composition) Let A and B be two OPNs, their composition is an OPN $A \oplus B = (P, I, O, Y, F, i, f)$ defined by:

- (1) $P = P_A \cup P_B \cup (I_A \cap O_B) \cup (I_B \cap O_A)$
- (2) $I = (I_A \cap O_B) \cup (I_B \cap O_A)$

- (3) $O = (O_A \cap I_B) \cup (O_B \cap I_A)$
- (4) $T = T_A \cup T_B$
- (5) $F = F_A \cup F_B$
- (6) $i = i_A \cup i_B$
- (7) $f = f_A \cup f_B$

In the paper, we use other methods to solve the problem that the T-invariant does not exist. Based on the service tree and its properties, the paper proposes a decision algorithm for weak soundness of Web service composition. First we introduce some basic concepts of service tree.

Definition 6(Pairwise Composable) Let A_1, \dots, A_n be pairwise composable OPNs. Let $c : \{2, \dots, n\} \rightarrow \{1, \dots, n-1\}$ be such that:

$$\begin{aligned} \forall i \in \{2, \dots, n\} : c(i) < i, \\ \forall i \leq i < j \leq n : i = c(j) \Rightarrow \\ I_{A_i} \cap O_{A_j} \neq \emptyset \vee O_{A_i} \cap I_{A_j} \neq \emptyset, \\ \forall i \leq i < j \leq n : i \neq c(j) \Rightarrow \\ I_{A_i} \cap O_{A_j} = \emptyset \vee O_{A_i} \cap I_{A_j} = \emptyset. \end{aligned}$$

Definition 7[14] (Condition) Let A and B be two composable OPNs. Condition $\Omega_{A,B}$ holds if and only if

$$\begin{aligned} \forall m \in R(A \oplus B), \sigma \in (T_A)^* : (A : m \mid P_A \xrightarrow{\sigma} f) \\ \Rightarrow (\exists \tilde{\sigma} \in (T_A \cup T_B)^* : (A \oplus B : m \xrightarrow{\tilde{\sigma}} f_A + f_B) \\ \wedge \tilde{\sigma} \mid T_A = \sigma) \end{aligned}$$

Here, T^* is represented as the closure of T , $P \xrightarrow{\sigma} f$ is represented as the P reach to f though the transition sequences σ .

Theorem 1 Let A_1, A_2, \dots, A_n be a service tree with root A_1 and tree function c . Further, let A_1 be weak sound and for $2 \leq i \leq n$, $\Pi_{A_i, A_{c(i)}}$ holds. Then $A_1 \oplus \dots \oplus A_n$ is weak sound.

Based on the theory, we give the weak soundness decision algorithm.

Algorithm 1 Weak soundness

Require: A_1, \dots, A_n are service components.

Ensure: Decision Result.

- 1: First, the components are modeled by OPNs. Then, if A_1, \dots, A_n are pairwise composable, go into (2); else algorithm terminates;
- 2: If A_1 satisfies weak soundness, then continue step (3); else algorithm terminates;
- 3: Determine whether A_1, \dots, A_n can form service tree, i.e. $c : \{2, \dots, n\} \rightarrow \{1, \dots, n-1\}$, whether c satisfies the three conditions: $\forall i \in \{2, \dots, n\} : c(i) < i$, $\forall i \leq i < j \leq n : i = c(j) \Rightarrow I_{A_i} \cap O_{A_j} \neq \emptyset \vee O_{A_i} \cap I_{A_j} \neq \emptyset$ and $\forall i \leq i < j \leq n : i \neq c(j) \Rightarrow I_{A_i} \cap O_{A_j} = \emptyset \vee O_{A_i} \cap I_{A_j} = \emptyset$.
- 4: Based on Theorem 1 determine whether when $2 \leq i \leq n$, $A_i + A_{c(i)} \geq A_{c(i)}$ holds. If so, then continue step (5); else algorithm terminates;
- 5: $A_1 \oplus \dots \oplus A_n$ are weakly sound and algorithm terminates.

3 The behavior conformance of composite web service under the open environment

3.1 Analyzing the dependent relationships of the composite web service

In view of the behavior dependent relationships[15,16,17,18,19] between transitions, the behavior dependent relationships include two kinds: direct behavior dependent relationship and indirect behavior dependent relationship. To the direct behavior dependent relationship, the analysis is very simple, the corresponding process mining method is easy to implement. But, the indirect behavior dependent relationship is difficult to tackle, so this need take some measures to mine their behavior model[20].

Definition 8[21] (Basic order relationship) Let W is a event log, $N = (P, T; F)$ is the building Petri net of business process, here, $W = L(N)$, $a, b \in T$

- (1) $a >_w b$ iff $\exists \sigma = t_1 t_2 t_3 \dots t_n \in W$,
 $i \in \{i, \dots, n-1\} : t_i = a \wedge t_{i+1} = b$;
- (2) $a \Delta_w b$ iff $\exists \sigma = t_1 t_2 t_3 \dots t_n \in W$,
 $i \in \{i, \dots, n-2\} : t_i = t_{i+2} = a \wedge t_{i+1} = b$;
- (3) $a \diamond_w b$ iff $a \Delta_w b \wedge b \Delta_w a$;
- (4) $a \rightarrow_w b$ iff $a >_w b \wedge (\neg(b >_w a) \vee (a \diamond_w b))$;
- (5) $a \#_w b$ iff $\neg(a >_w b) \wedge \neg(b >_w a)$;
- (6) $a \parallel_w b$ iff $a >_w b \wedge b >_w a \wedge \neg(a \diamond_w b)$;

As can be seen from the definition 1, $>_w$ and Δ_w are the basic order relationship, the former represents the two tasks occur one after another, the latter represents two tasks can generate a specific piece of the cycle track. These can be used to difference the relationships of two tasks. (3) (4) (5) (6) corresponding to cycle, sequence, select, and concurrency relations respectively.

Definition 9[21] (Direct dependent relationship) $N = (P, T; F)$ is the building Petri net of business process, for arbitrary $a, b \in T$, if there exists direct dependent relationship between a and b iff

- (1) $a \bullet \cup b \bullet \neq \Phi$
- (2) There exists reachability mark $s \in [N, [i] >$ to make $(N, s)[a >$ and $(N, s - \bullet a + a \bullet)[b >$

Definition 10[21] (Indirect dependent relationship) $N = (P, T; F)$ is the building Petri net of business process, for arbitrary $a, b \in T$, if there exists indirect dependent relationship between a and b iff

- (1) $a \bullet \cap b \bullet \neq \Phi$ and arbitrary $p \in a \bullet \cap b \bullet$, p is not the implicit place;
- (2) There do not exist reachability mark $s \in [N, [i] >$, to make $(N, s)[a >$ and $(N, s - \bullet a + a \bullet)[b >$;
- (3) There exists reachability mark $s \in [N, [i] >$, to make $(N, s)[a >$ and exists reachability mark $s' \in [N, s - \bullet a + a \bullet >$, to make $(N, s')[b >$.

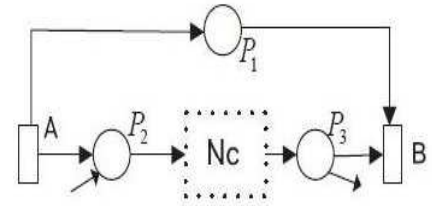


Fig. 1 The basic structure of process model with indirect behavior dependent relationships

Next, we analyze the architectural feature of the process model with the indirect behavior dependent relationships. Suppose there exist indirect behavior dependent relationship between the task A and the task B, after the task A executes, there must have other tasks executing before the task B, then the task B can be executed. So, the task B and the task A never appear one after another in the log traces, and we have not good methods to check the indirect behavior dependent relationships. The basic structure of process model with indirect behavior dependent relationships is showed in the Figure 1.

In the Figure 1, the subnet N_c includes one task at the least, it is a transition-type subnet. Generally speaking, subnet N_c may be have complex behavior relationships with the other sections of the process model. Here, we only consider the simplified situation, some complex situation can be transformed into the simplified situation by reduction methods. The extended structure of process model with indirect behavior dependent relationships is show in Figure 2.

In the Figure 2(a), it represents the first situation, the place P_1 has not the input and output, but the P_2 has the input, the P_3 has output. For the first situation, the task A and the task B have the indirect behavior dependent relationship, the behavior expression is as follows:

$$\neg(A > B) \wedge \exists t \in T, p_1, p_2 \in P : (p_1 \in \bullet \wedge p_2 \in \bullet t \wedge A \in \bullet p_1 \wedge A \notin \bullet p_2 \wedge B \in p_2 \bullet \wedge \neg \exists A' \in \bullet p_2 : (A' \succ_w A \vee A' \parallel_w A))$$

In the Figure 2(b),(c), it represents the second situation, the place P_1 has the input and the output, but only one, the input and the output can not exist at the same time. P_2 has zero or one input, P_3 has zero or one output. For the first situation, the task A and the task B have the indirect behavior dependent relationship, the

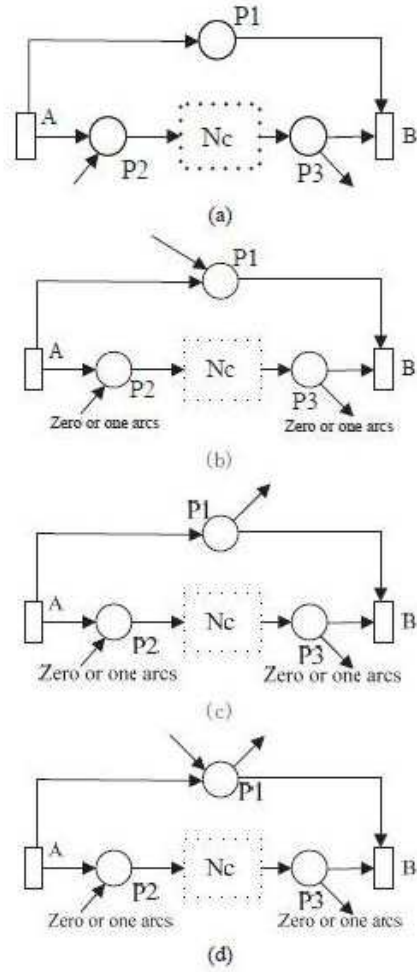


Fig. 2 The extended structure of process model with indirect behavior dependent relationships

behavior expression[16] is as follows:

$$\begin{aligned}
 & (A \gg_w B \wedge |A^*| > 1 \wedge \exists B' \in T : B \triangleleft_w B' \wedge \\
 & \exists p \in A^* : (\neg \exists t \in p^* : (t \succ_w B \vee t \parallel_w B)) \wedge \\
 & \exists t' \in p^\bullet : (t' \succ_w B' \vee t' \parallel_w B')))) \\
 & \vee (A \gg_w B \wedge |B| > 1 \wedge \exists A' \in T : A \triangleright_w A' \wedge \\
 & \exists p \in B^\bullet : (\neg \exists t \in p^\bullet : (A \succ_w t \vee A \parallel_w t)) \wedge \\
 & \exists t' \in p : (A' \succ_w t' \vee A' \parallel_w t'))).
 \end{aligned}$$

In the Figure 2(d), it represents the third situation, the place P1 has the input and output, P2 has zero or one input, P3 has zero or one output. For the third situation, the task A and the task B have the indirect behavior dependent relationship, the behavior expression is as

follows:

$$\begin{aligned}
 & \exists A', B' \in T : (A^\bullet \cup A^\bullet \neq \Phi \wedge A \gg_w B \wedge \neg(A \gg_w B')) \\
 & \wedge A' \gg_w B' \wedge \neg(A' \gg_w B) \wedge B^\bullet \subseteq B' \cup \{t \mid t \neq (A \gg_w t) \\
 & \wedge A' \gg_w t \wedge (B' \parallel_w t \wedge B' \succ_w t) \wedge B^\bullet \wedge t \neq \Phi\}.
 \end{aligned}$$

3.2 Building K-bounded process behavior model based on the new process mining methods

According to the problem characteristic, the Web service process mining is important. If we adopt the existing mining methods, it is difficult to mine the exact model which is behavior conformance with the theoretical composite model. Since each mining algorithm has its limitations, it is necessary to find new methods to solve this problem. For example, a-series algorithms can only mine 1-bounded nets, but the composite Web service process may not be 1-bounded. The region-based mining methods can mine K-bounded ($k > 1$) model, but it is difficult to deal with the behavior dependent relationships.

Due to the particularity of the problem, we analyze the behavior conformance between the mined models with the theoretical composite model. In the analysis process, we can obtain some behavior dependent information and K-bounded value by analyzing the theoretical composite model in advance, which can induced the Web service process mining.

Algorithm 2 K-bounded process mining algorithm based on induction information

Require: log traces, induction behavior dependent relationships.

Ensure: the Petri net model of Web service process.

- 1: Transforming the log traces into the corresponding transition system, then analyzing the direct dependent relationships between all tasks (or transition);
- 2: Let $TS = (S, E, A, s_{in})$, $PN = (P, T; F, M_0)$, for each $r \in R$, each r is corresponding to a place p , $p_i = r_i$, $|p| = |r|$;
- 3: For each $r \in R$, $M_0[r] = r(s_{in})$, each e is corresponding to a place t , $t_i = e_i$, $|t| = |e|$;
- 4: Let $ER(e) = \{s \mid \exists s' : (s, e, s') \in E\}$, $SR(e) = \{s \mid \exists s' : (s', e, s) \in E\}$, if $ER(e) \subseteq r$, then linking a directed arc from the p which corresponding to r to t which corresponding to e . if $SR(e) \subseteq r$, then linking a directed arc from the t which corresponding to e to the p which corresponding to r . So, a Petri net model can be obtained, and the model is called as λ_0 ;
- 5: Using the induction behavior dependent relationships, selecting one of dependent relationship to adjust the model λ_0 , and obtaining the new model λ .
- 6: Computing the metrics value of model λ_0 and model λ based on the log sequences, if $(fitness(\lambda, L) \leq fitness(\lambda_1, L) \leq fitness(\lambda_0, L))$ and $precision(\lambda, \lambda_0, L) \leq precision(\lambda_1, \lambda_0, L)$ or $(fitness(\lambda, L) \leq fitness(\lambda_1, L) \leq fitness(\lambda_0, L))$ and $recall(\lambda, \lambda_0, L) \leq recall(\lambda_1, \lambda_0, L)$, then $\lambda_0 = \lambda$, $\lambda_1 = \lambda$; else, the initial model λ_0 keeps unchanging, then considering the next induction information;
- 7: If the induction information has been learned end or the model λ is not change, then the modeling process end, and output the model λ .

Table 1 The results of two methods

Index \ Method	Interacting BP	(BP ₁ ,BP ₂)	(BP ₁ ,BP ₃)	(BP ₁ ,BP ₄)	(BP ₁ ,BP ₅)	(BP ₂ ,BP ₃)
	K-bounded	1	1	2	2	2
Our Method	K-bounded	1	1	2	2	2
	Structure Fitness	1	1	1	1	0.996
	Behavior Precision	1	1	0.985	0.978	0.971
	Behavior Recall	0.992	0.986	0.978	0.966	0.959
Genetic Method	K-bounded	1	1	1	1	1
	Structure Fitness	1	1	0.993	0.989	0.984
	Behavior Precision	0.996	0.994	0.975	0.958	0.920
	Behavior Recall	0.969	0.971	0.937	0.915	0.895

4 Simulation experiment

Based on the methods proposed in the paper, we use the ProM framework to analyze the behavior trustworthiness of component interaction. The ProM framework[22] is an open-source tool and it can be downloaded at www.processmining.org, specially tailored to support the development of process mining plug-ins.

Test. For the component cases from the large Benchmarks[23], giving some known induction information, such as the K-bounded value and some dependent relationships, using our methods and genetic methods presented in the literature[12] respectively, we compare the some indexes using two methods under different interacting Web service process.

In Table 1, a tuple (Bp_i, Bp_j) represents the interaction between the Web service Bp_i and the Web service process Bp_j. By analyzing, the K-bounded value is given. The experiment results show that our methods are better than the genetic mining methods, especially, when the K-bounded value of composite Web service process is larger than 1, the difference is larger.

5 Conclusion

At present, the behavior trustworthiness of composite Web service process is a study focus, the essence characteristics of trustworthy Web service process are the execution result and behavior can be predictable. Under the open and dynamic changing environment, some complex demands need several Web service processes interacting to implement, these cause the composite Web service process to behave uncontrollably, uncertainly, and unpredictably. The paper presents a behavior trustworthiness analysis method of Web service process based on induction information. Firstly, aimed to the internal factors, we analyze the behavior weak soundness

in order to guarantee the predictable function of Web service process. Then, for the external factors, in order to analyze the behavior change of Web service process, we propose a new process mining methods. Finally, using methods above mentioned, we test our methods in large Benchmark, and compare our methods with genetic process mining. Theoretical analysis and experimental results indicate that our method is better than the methods of genetic process mining.

Based on the theoretical analysis and experimental results, the innovation and advantage of the paper are: 1) In order to analyze the predictable interaction function, behavior weak soundness based on Petri net is presented. 2) Web service process mining methods with the indirect behavior dependent relationships is proposed, which can take into account behavior relationships from the theoretical composite model, and avoid the blindness of building process model.

In the future, we would like to study the trustworthy evaluation of networked software, and study the adaptation methods of non-consistent behavior relativity. Moreover, it is also one of our future works to study the intelligence and dynamic behavior analyzing methods of complex Web service process.

Acknowledgement

We would like to thank the support of the National Natural Science Foundation of China under Grant No.61272153, No.61170059, No.61100058, No.60973050, 61170172 and No.71071003, the National High-Tech Research and Development Plan of China under Grant No.2009AA01Z401, the Natural Science Foundation of Educational Government of Anhui Province of China (KJ2012A073, KJ2011A086), Anhui Provincial Natural Science Foundation (1208085MF105), Anhui Provincial Soft Science Foundation

(12020503031), Excellent Talent Project of Anhui Province of China (2009SQRS 045), Social Science Project of Anhui Province of China (2009SK156).

References

- [1] Z. M. Zheng and S. L. Ma, Science in China Series F-Information Science, **8**, 1328 (2009).
- [2] X. W. Fang, C. J. Jiang and X. Q. Fan, International Journal of Computational Intelligence Systems, **5**, 542 (2010).
- [3] A. K. Alves de Medeiros, W. M. P. van and A. J. M. M. Weijters, Data & knowledge engineering, **64**, 55 (2008).
- [4] H. Foster, S. Uchitel, J. Magee and J. Kramer, Proc. of the Int'l Conf. on Web Service (ICWS), 738 (2004).
- [5] X. Fu, T. Bultan and J. W. Su, Proc. Of the Int'l World Wide Web Conf.(WWW), 621 (2004).
- [6] A. Martens, Petri Net Newsletter, **65**, 12 (2005).
- [7] Y. Y. Yin, Y. Li, S. G. Deng and J. W. Yin, Acta Electronica Sinica, **37**, 433 (2009).
- [8] X. W. Fang, C. J. Jiang, Z. X. Yin and X. Q. Fan, Computer Science and Information Systems, **8**, 843 (2011).
- [9] G. Ammons, R. Bodik and J. R. Larus, Proc. of the 29th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, **37**, 1 (2002).
- [10] R. Bergenthum, J. Desel and R. Lorenz, BPM 2007 of LNCS 4714, 375 (2007).
- [11] J. Carmona, J. Cortadella and M. Kishinevsky, IEEE Transactions on Computers, **3**, 371 (2010).
- [12] A. MEDEIROS, A. WEIJTERS and V.D.AALST, In Workshop on Business Process Intelligence, Lecture Notes in Computer Science, **3812**, 203 (2006).
- [13] N. Lohmann, P. Massuthe and K. Wolf, Computer Science, **1**, 321 (2007).
- [14] W. M. Aalst, K. M. Hee, P. Massuthe, N. Sidorova and J. M. Werf, Proceedings of the 30th International Conference on Applications and Theory of Petri Nets, 283 (2009).
- [15] M. P. Das, Mesoscopic systems in the quantum realm: fundamental science and applications, Advances in Natural Sciences: Nanoscience and Nanotechnology, **1**, (2010).
- [16] T. Shahwan, R. Said, A Comparison of Bayesian Methods and Artificial Neural Networks for Forecasting Chaotic Financial Time Series, Journal of Statistics Applications & Probability, **1**, 89-100 (2012).
- [17] K. S. Mwitondi, R. A. T. Said, A data-based method for harmonising heterogeneous data modelling techniques across data mining applications, Journal of Statistics Applications and Probability, (In Press).
- [18] A. Chand, A heuristic approach to constraint optimization in timetabling, The South Pacific Journal of Natural Science, **20**, 64-67 (2002).
- [19] H. E. Maroufy and Z. Taib, Applied Mathematics & Information Sciences, **4**, 353(2010).
- [20] W. M. P. van der Aalst and C. Ouyang, ACM Transactions on Internet Technology, **3**, 13 (2008).
- [21] Wen Lijie, Wil M. P. van der Aalst and Wang Jianmin, Data Mining and Knowledge Discovery, **2**, 145 (2007).
- [22] A. J. M. M. Weijters, W. M. P. van der Aalst, A. K. Alves de Medeiros, BETA Working Paper Series, WP, **166**, (2006).
- [23] A. Rozinat and W. M. P. van der Aalst, Workshop Business Process Intelligence (BPI '05), 1 (2005).



Xiangwei Liu

received her M.A. degree from Anhui University of Finance and Economics, China, in 2005. She is currently an associate professor with the Department of Computer Science and Engineering, Anhui University of Science and Technology, China. Her

current areas of research are Web service computing, Petri net and formal verification of software. He has published more than 20 papers in the international academic journals, these papers are embodied more than 20 times by SCI and EI and are cited more than 60 times by others.



Xianwen Fang

received his M.A. degree from Shandong University of Science and Technology, China, in 2004, and PhD. degree in the key Lab of Service Computing at Tongji University in 2011. He is currently a Professor with the Department of Computer Science and Engineering,

Anhui University of Science and Technology, China. His research interests include Petri net, trustworthy software and Web services. He has published more than 60 papers in domestic and international academic journals and conference proceedings. These papers are embodied more than 40 times by SCI and EI and are cited more than 100 times by others.



Mimi Wang

received the M.A. degree from Anhui University of Science and Technology, China, in 2011. She is currently a lecturer with the Department of Computer Science and Engineering, Anhui University of Science and Technology, China. Her

current areas of research are concurrent theory, Petri net and formal verification of software.

**Zhixiang Yin**

received
her M.A. degree from
Nanjing Normal University
of China in 1996, and
PhD. degree in the Huazhong
University of Science
and Technology in 2003.
His research interests include
Petri net and trusted software.