ATLANTIS PRESS International Journal of Networked and Distributed Computing, Vol. 5, No. 3 (July 2017) 152–163

Research on Service Recommendation Reliability in mobile computing

Weng Wen[1,2], Huaikou Miao [3]

[1] *School of Computer Engineering and Science, Shanghai University,*
Shanghai Key Laboratory of Computer Software Evaluating & Testing,
99 Shangda Road, BaoShan District
Shanghai,200444, China
E-mail: wengwen@shu.edu.cn

[2] *College of Computer and Information Engineering, Shanghai Polytechnic University*
2360 JinHai Road, PuDong District
Shanghai,201209, China
E-mail: wengwen@shu.edu.cn

[3] *School of Computer Engineering and Science, Shanghai University,*
Shanghai Key Laboratory of Computer Software Evaluating & Testing,
99 Shangda Road, BaoShan District
Shanghai,200444, China
E-mail: hkmiao@shu.edu.cn

Abstract

Web services bring more conveniences for users and developers. However, it makes user face the problem of service information explosion. The personalized service recommendation solves the problem. This paper proposes a mechanism to evaluate Web service composition's reliability which bases on user context information in mobile computing. The mechanism constructs user behavior model by formatting user context and then quantitative verification is performed to estimating whether the recommended Web service composition satisfy the requirements of users. Finally, experiments are carried out to demonstrate the effectiveness of our mechanism.

Keywords: Context-aware; Service recommendation; Probability model; Mobile computing; Model checking.

1. Introduction

With the development of mobile communication, wireless network, database, distributed computing, Mobile computing technology will enable the intelligent terminals to achieve data transmission and resource sharing via wireless networks. Mobile devices are increasingly becoming an indispensable part of human life. A large number of the Web applications on the PC and the Internet have been moved to the mobile platform. But it's difficult for user to choose a satisfying one from massive Web services with similar functions. Personalized service recommendations solve the problem in an active way [1]. Due to the characteristics of the mobile computing environment, a variety of mobile devices can be more accurately obtain user context information to infer user's real-time needs, user-centered and featured with the on-demand intelligent recommending services through service dynamic synthesis is becoming the focus of research [2]. However, the limited resources of mobile devices and wireless network, such as battery life, storage, bandwidth and mobility, any abnormalities may immediately influence the correctness and reliability of service recommendation and seriously impact the user experience and core business process of service-oriented software system [2]. At present, most of the context-based recommended technology concern on recommending a set of functional correct Web services [3] and few studies provide relevant strategies to dynamically evaluate the functional and nonfunctional attributes of Web service recommendation in mobile computing environment. Functional correct service refers to the ability to correctly meet users needs of the services. And the non-functionality of the service refers to the security, reliability, and availability of the services, that is, the degree to which the service correctly performs the task to meet the user.

Therefore , how to effectively evaluate the set of recommended services to meet users' functional and non-functional needs is a big challenge.

Most of the research focus on the use of formal verification method for the correctness of Web services. Paper[4] addresses the issue of verifying composite Web services by using the model checker NuSMV. Diaz et al. [5] proposed a method to generate correct WS-BPEL skeleton documents from WS-CDL documents by using the Timed Automata as an intermediary model in order to check the correctness of the generated Web Services with model checking techniques. Belli F et al. [10] proposed a new model to verify the correctness of web service compositions. It can be seen that model checking technology is widely used and gives qualitative results in the form of "yes" or "no". More and more researchers are concerned with quantitative results, and they paid more attention to whether the quality of the service satisfy uses' needs. For instance, Quan Z.Sheng[6] proposed an automated service verification approach to verifies the properties in operational behaviors using the NuSMV model checker. Guoxin Su et al. [7] addressed the reliability of QoS evaluation using parametric model checking. G Babin et al. [8] addressed the problem of the correct design of Web service compositions in case of failures and verifies the service function through the model checking techniques. Moreno G A et al. [9] presented an approach for proactive latency-aware adaptation under uncertainty that uses probabilistic model checking for adaptation decisions. Chen M et al. [11] proposed an automated method of directly verifying the combined functionality and nonfunctional requirements through the Web service composition semantics. All of the above studies are based on Web service itself, to measure and evaluate the functionality and quality of Web service.

In this paper, we propose a mechanism to evaluate the reliability of recommended Web service composition in a mobile computing environment, which takes full account of mobile users' context information and then verify the quality of the recommended Web service composition by probability model technology.

An real-time state of a mobile user can be described by a number of contexts, such as weather, time, location, movement speed, habits, preferences, etc. Assuming a state corresponds to a class of demands, user different demands correspond to different states, when user switches between different states, we construct user behavior model by way of migration probability matrix to describe user state transformation. The probability calculation tree logical (PCTL) can be used to describe system requirement property, and the probabilistic model verifier PRISM can be employed to check the requirement property for estimating whether the recommended Web services meet the requirements of users. The probability value obtained by the model checker is evaluated by the mathematical statistical analysis method to judge how the recommended service meets the user's needs. At the same time, the evaluation result will be used as the basis for the next service recommendation. Thus, we can implement to verify correctness of the recommended Web service composition and reliability estimation of recommened Web services. As shown in Figure1.

Our service recommendation mechanism in mobile computing includes three steps.

(i) Through mobile devices and sensors, collecting user context, filtering invalid context information and formatting valid context information effectively, and then determining the user requirements.

(ii) Based on the user requirements identified in the previous step, combined with business processes and user choice history, Web services is recommended to user.

(iii) Probabilistic model varivifation is used to evaluate the recommended Web services. In the evaluation results, those services highly matching the user's requirements will be recorded.

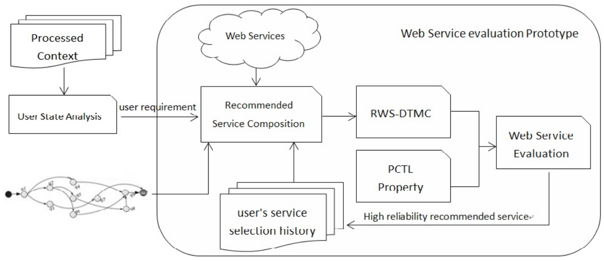

Fig. 1. Web service evaluation prototype

This paper is organized as follows: Section 2 introduces main techniques. Section 3 proposes the reliability evaluation mechanism of our apporch. Section 4 introduces probabilistic model basis. Section 5 shows a case study and experiments to verify our method. Section 6 draws a conclusion and future research directions.

2. Main Techniques

2.1. Web Services Composition

Web service has been an important solution to achieve resource sharing and application integration in the Internet era. With the advent of the cloud computing, Web services in the cloud bring more conveniences for users and developers. However, it makes user faced the problem of service information explosion. It's difficult for user to choose a satisfying one from many Web services with similar functions. Personalized service recommendations solve the problem in an active way. Under the interoperability of Web service, complex business interactions fulfilled by the process of dynamic discovery, integration, coordination and execution of distributed service entities (atomic or composite service), via published and discoverable interfaces, can satisfy

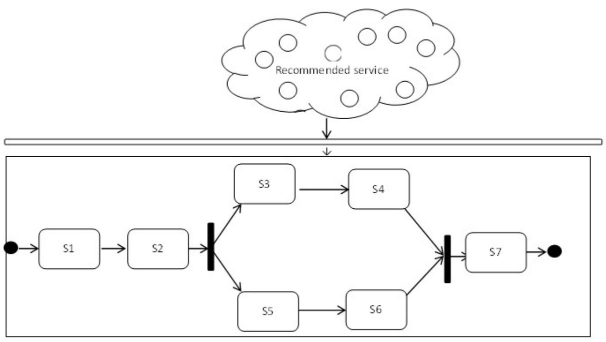

Fig. 2. Each service recommendation in the service composition

the needs and interests of mobile users in real time. BPEL4WS,WS-CDL, OWL-S and other languages have been developed to Orchestration and Choreography Web services based on business processes. Each node in Figure 2 has many same function Web service recommended by personalized service recommended method.

2.2. Service recommendation technology

The issue of service recommendation have been discussing since the mid - 1990 s [1]. It has gradually developed into an independent research field. Personalized recommendation solutions have been improved and implemented by the Academia and Industry, and have been widely used in e-commerce platform. Traditional recommendation mechanisms commonly use collaborative filtering [12]. CF algorithms assume that in order to recommend items to users, first of all , evaluate the level of the item and then the item obtains the highest number of votes would be recommended to similar users. The collaborative filtering recommendation algorithm based on the Nearest Neighbor idea is mainly divided into two categories: user-based [13] and project-based [14]. Another branch of the collaborative filtering algorithm is model-based, such as Bayesian network model [15], neural network model [16], probabilistic model [17] and so on. The latter more considers the influence of the sparseness of the matrix on the recommended results in the actual situation while the number of users, items and services become large and large. Therefore, the CF algorithm based on model mainly use machine learning methods, through a large number of samples of training, to build a service requirements model, as the basis for the recommendation. In the model-based service recommendation model, combined with the user context, this personalized recommendation, the type of personalized service recommendation will enhance the accuracy of service recommendations and improve the experience of mobile users. Figure 3 shows the classification of the CF algorithm.

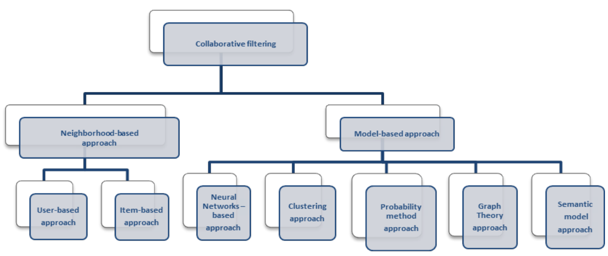

Fig. 3. Classification of Service Recommendation Approaches

2.3. Context and Context-Aware

With the rapid development of the mobile device (wearable equipment, smart phones) today, service recommendation technology is also increasingly intelligent. In the mobile computing environment, the recommendation technology will be more reference to the users' context information obtained by all kinds of mobile devices. Context-based recommendation technology can be fully user-centric and allow user to use Web service resources on demand.

The concept of context-aware was first proposed by Schilit and Theimer in the early nineties [18]. The context information is divided into three categories [19]: User context, which contains personal information and preferences, often go to places, can also be users of instant temperature, pulse, heart rate and other information. Physical context can be position, temperature, humidity, environmental noise and traffic information. Time context refers to a period of time, one day, a week, a month or a season. etc. [19]. In the three ways [19] to get the context, direct access to the context is the most straightforward and accurate. At the same time , it is very important to obtain the contextual information by implicit and reasoning methods.

We generate a lot of context information at all times, such as heartbeat, body temperature, location, movement speed, etc., which can be collected by mobile devices and be used to provide us with intelligent services. Mobile devices already have the perceive ability which traditional personal computers do not have. Mobile devices are able to perceive a dynamically changing context. Research on context and context-aware is also becoming more and more popular. For instance, the geographical location of the mobile user can be detected by the increasingly sophisticated mobile data network and positioning system [20]. Tour guide service [21] and commercial recommendation service [22] are recommended by using the user's contextual information (location, ID and time). Paper[29,30] are through the mobile device to perceive the user location context to achieve the user's journey planning and tour guide. Since wireless information access is now widely available, there is a high demand for accurate positioning in wireless networks, including indoor and outdoor environments. If mobile user is outdoors, the context information about location can be easily obtained by GPS. However, if he is indoors, getting his location context is not so easy. Paper[31,32] proposed the methods about obtaining indoor user's location context through the wireless indoor positioning system.

Not all of the context information is useful and context sensed from different mobile devices and sensors may conflict with each other. Useless context information must be filtered out and valid context information also needs to be translated into a uniform format. Among these standards, resource Description Framework (RDF) provides data model specifications and XML-based serialization syntax, Web Ontology Language OWL) [16] enables the definition of domain ontologies and sharing of domain vocabularies. OWL is modeled through an object-oriented approach, and the structure of a domain is described in terms of classes and properties,from a formal point of view.

```
……
< Class  ID=" ContextEntity" />
< Class  ID=" Location" >
  < subclassOf   resource=" #ContextEntity" />
</Class>
</Class>
< ObjectProperty  ID=" longtitude" >
  < type resource=" FunctionalProperty" >
  <domain resource=" Location" >
  < range resource=" double" >
< /ObjectProperty>……
< Class ID=" Indoor" >
  < subClassOf  resource=" #Location" / >
  < disjointWith resource=" #Outdoor" />
</Class>
……
```

Fig. 4. Partial OWL serialization of the upper ontology.

The structure and properties of the context can be described by ontology. Common ontology language is an object-oriented language which described a domain by classes and attributes [16]. The defined form of expression is (subject, predicate, object), where the objects of the ontology is expressed by subject and object, the predicate means the attribute relation.

2.4. Response Time Contraints

Taking into account the equipment performance and resource saving, context information acquisition time interval can be determined by the heartbeat mechanism whose cycle is not fixed [19]. So the user's context information is detected and updated once in a while, we can get the user's contextual information in real time to infer the user's state. [25]. Dynamically changing formatted context information can form a sequence. We agreed to get the user context information from mobile devices every 10 seconds. By mean of a set of formatted context information to determine the user's requirements which correspond to a Web service with similar functions.

On the other hand, As the mobile device screen is small, the user will be staring at the screen waiting for a service response. Once the user's status changes, the system will recommend the service to the user within the specified time which is called "response time". The length of the response time directly affects the user's experience. In general, there is a standard of response time. It called “2/5/10 seconds principle”. That is, It will be considered "very attractive" user experience if user is responded within 2 seconds. Within 5 seconds to respond to the user is considered "relatively good" user experience and within 10 seconds to the user response is considered "bad" user experience. If more than 10 seconds have not been answered, then most users will think that this request is a failure. So the service response time and cost is what we should consider.

Table 1. The correspondence between response timea and user experience.

ID	Response Time(s)	User Experience
1	0 < Response Time <=2	very attractive
2	2 < Response Time <=5	relatively good
3	5 < Response Time <=10	bad
4	Response Time>10	failure

3. Reliability Evaluation Mechanism

3.1. Unitary linear equation

The traditional way user interacts with Web service is the request-response mode. When user makes a request, server responses result to user, such as the login system. The context-based service recommendation model is based on user context information to infer the user's current state. Our another paper[23] proposes a context-based service intelligence recommendation mechanism. Formats the collected valid context information, and describe them by OWL. The hidden Markov model is established in terms of user states and context information is serialized as a parameter input to the model. Context determines user's current state. The Baum-Welch algorithm of the hidden Markov model is adopted to infer users’ next state and user is recommended for Web services meeting his needs. In that mechanism, user need not explicitly request a service access, but rather the system infers the user state based on the contextual information obtained by the mobile device.

Faults may be occurs during service recommendation since bandwidth fluctuations, connection instability and too long wait time. These faults are caused by the characteristics of the mobile computing. In addition to improving hardware, it is an effective way to improve the accuracy of service recommendations by pre-evaluating the reliability of services and keeping records of high reliability services that have been evaluated.

The reliability analysis of Web service combination is an analysis of the reliability of existing Web services portfolio, predicting future service reliability trends, and saving the analysis results to the historical records. We use the linear regression analysis proposed by Gao et al [26]. To evaluate the reliability of the recommended service. In the service reliability analysis process, the reliability of the recommended Web service behavior is evaluated by analyzing the probability model test data.

In this paper, the probabilistic results obtained by probabilistic model are used as sample data, and the quantitative dependence between reliability of recommended Web service and constraint time is simulated by linear regression method. The results obtained by the linear regression equation are saved to historical records as a basis for the next recommendation.

In this paper, we use the following linear regression equation to evaluate the reliability of recommended services, that is, the probability that a set of services is successfully recommended.

$$y = a + bt \tag{1}$$

The symbol t represents the services response time in the mobile computing environment, the symbol y represents the reliability probability value of recommended services, a and b are the coefficients of the linear regression equation. The reliability probability value varies with the required response time t.

To establish this linear equation, we can easily determine the value of a and b, but how to make them become the linear equation model optimization parameters, least squares is a mature and efficient choice which ensure the sum of the squares of all data deviations is minimal .

Assume that the sum of squares of all data is M.

$$M = \sum_{i=0}^{n}[y_i - (a + bt_i)]^2 \tag{2}$$

yi and ti are known, then the equation is transformed into a binary function with a and b as independent variables, M as the dependent variable. By computing partial derivatives of M, we get the following equations.

$$\begin{cases} \dfrac{\partial M}{\partial a} = -2\sum_{i=0}^{n} t_i[y_i - (a + bt_i)] = 0 \\ \dfrac{\partial M}{\partial b} = -2\sum_{i=0}^{n}[y_i - (a + bt_i)] = 0 \end{cases} \tag{3}$$

After further calculations, the next steps and results are available.

$$\Rightarrow \begin{cases} \text{na} + \text{b}\sum t = \sum y \\ \text{a}\sum t + b\sum t^2 = \sum ty \end{cases} \quad (4)$$

$$\Rightarrow \begin{cases} b = \dfrac{n\sum ty - \sum t \sum y}{n\sum t^2 - (\sum t)^2} \\ a = \dfrac{\sum y}{n} - b\dfrac{\sum t}{n} \end{cases} \quad (5)$$

After a series of derivation, we can get the values of a and b. After the regression coefficient is obtained, the evaluation of the fitting degree and the standard error according to the linear regression equation is the reliability evaluation of the recommended services.

3.2. Probabilistic model modeling basis

Definition 1 (RWS-DTMC). The user behavior model is a finite state machine, that describes changes of user's status. It is a tuple (S, S0, R, AP, L), where,

－S is a non-empty finite set of states; Each state represents the Web service entity recommended to the user

－S0 is an initial state; $S0 \in S$

－R: S $\times$S is a finite state migration set; S $\times$S ->[0,1] for all $s \in S$

$$\sum_{s' \in S} P(s, s') = 1 \quad (6)$$

－AP is a finite set of atomic propositions;

－L : S → 2 AP is a proposition assignment function. It describe the properties of the service.

RWS-DTMC focuses on probability reachability property, the quantitative verification can verify both functional and non-functional behavior. For simplicity, we define a unique initial state. Function of P, the transition probability matrix, for any state, $s \in S$, the probability P (s,s') refers to taking a transition from a state s to another state s', forming a discrete probability distribution for each state.

A path of a RWS-DTMC is a finite sequence Ɛ= s0,s1 . . . of states $si \in S$ such that P(si,si+1)>0 for all i. Let Paths(M, s) denote the set of all finite paths of M starting in s.We say that a state s' is reachable from another state s if there is a finite path from s to s'.

The probabilities of the cylinder sets are given by

$$P_r(\mathcal{E}) = \prod_{i=0}^{n-1} P(S_i, S_{i+1}) \quad (7)$$

Probabilistic real time Computation Tree Logic (PCTL) [33] is used to describe the system's real-time and state transition probabilities and is an adaptation of CTL to probabilistic systems with the abstract syntax . Formulas in PCTL are established by atomic propositions, propositional logic connectives and operators to express time and probabilities.

Definition 2. PCTL syntax is defined as follows:

$\Phi ::= true\ |\alpha|\varphi \wedge \phi\ |\neg\varphi|P(\sim p)\ [\psi]$

$\sim \in \{<, \leqslant, >, \geqslant\}$ is a relational operator.

$\Psi ::= X\phi\ |\ \phi\ U\ \phi\ |\ \phi\ U\hat{}(\leqslant t)\ \phi \quad t \in N$

Each atomic proposition a is a PCTL formula.

$p \in [0,1]$

The probability operator P allows to express probability thresholds on the probability mass of paths satisfying a formula. In PCTL one can formulate properties like "an erroneous state will be reached with a probability less than 0.01".

Using PCTL to describe the reliability of composite services to be verified, mainly focus on two kinds of properties.

(i) P~p[true U (system_state = success)] means the probability from the initial state to the combined state of success whether to meet the threshold P.

For example, $P_{<0.3}$(F state = success) means that "with probability 0.3 or less, a success state will be reached eventually". It refers to compute the maximal or minimal reachability probability.

(ii) system_state = "a certain service incocation"=> P~p[true U (system_state = success)] means the probability of reaching a successful state from a state in serivce composition satisfies the threshold value P.

For example, $P_{=?}$(F state = success) means that "what is the probability that a success state will be arrived?". It refers to iteration computing of matrix-vector multiplication representing the probability value of n-step reachability

4. Verification of service recommendation process

The process of quantitative verification involves functional verification and performance analysis. We adopt probabilistic model checking techniques to verify the Web service recommended to get its quantitative results, that is the correctness of the recommended Web service, as well as qualitative results, that is its probability. Web services recommended evaluation mechanism prototype shown in Figure 1, which several important components are described as follows:

(a) Processed Context: Obtain user context from time to time through various mobile devices and sensors, filter out invalid contexts, extract valid information, and describe it with OWL.

(b) User State Analysis: A set of context information can determine a state of the user and an user state can infer the service which is the user requirments. The system will handle the processed context as a parameter input hidden Markov model, according to user's current state can infer user's next state, that is, the next state of the requirement.

(c) Recommended Service Composition: In this step, the system will recommend Web services to user based on user requirement, Recommended service can use many recommendation mechanism mentioned in section II. In the process of the recommendation, in addition to considering the user current requirement, will be combined with business process as well as user's historical choice for recommendation service. This part is responsible for the execution of Web service composition.

(d) RWS-DTMC: User choose a service from a number of recommended services. The choice directly affect the system inference on user next state. User's choice probability is the user state transition probability. So user behavior model can be defined according to user' state. We model user behavior using RWS-DTMC modeler which take into account both functional and non-functional aspects.

e) Web Service Evaluation: This part is the focus of our paper. In this section, we evaluate the reliability of the recommended service by model checker PRISM. We translated the RWS-DTMC model from the previous step into the PRISM language, allowing PRISM to automatically verify the probability value of recomended Web service composition to which the conditional constraint is added. In addition, the service with high verification result is stored in the historical record for the next recommendation.

5. A case study and experiments

A mobile user plans to go to work from home, mobile devices recommend the way to the user according to the time, weather and other contextual information. It is a self-drive, a taxi ride or public transport. When he chooses a certain way, the system will continue to recommend the services he needs based on his status. For example, when a user selects to drive to the destination, the system will recommend him navigation services, or entertainment services (listen to the news or listen to music) after he got on the car, or user chooses a parking service when he get to destination. If the user selects a taxi to the destination, the system will recommend him entertainment services or mail service. If the user chooses to take public transport, the system will still recommend services for him. Of course, none of these services are needed that is possible. We represent the user's state transformation by figure 5. State S_0 recommends user traffic mode according to the processed context information (Marked as state S_0). User chooses to drive to the destination (Marked as state S_1). When he get on the car, the system recommend navigation services (Marked as state S_4) for him. If user does not need navigation services, the system continue to recommend entertainment or news services (Marked as state S_5). When user arrives at the destination, the system recommends parking service (Marked as state S_7). If user chooses to public transport (Marked as state

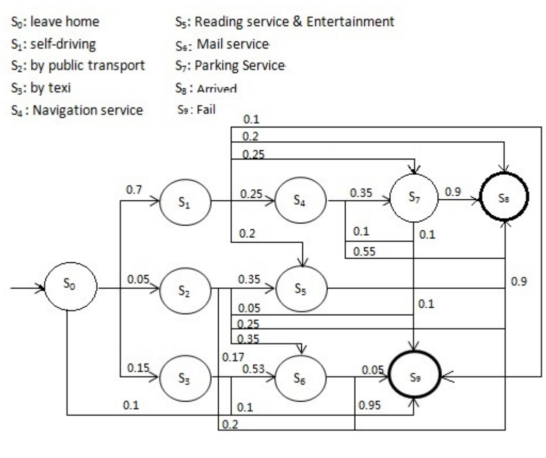

Fig. 5. User's state transition

S_2) or take a taxi (Marked as state S_3), the system recommend him mail service (Marked as state S_6). User also can choose none until State S8, the end state. The randomness of user behavior and the instability of mobile computing network could make the service recommendation failure, which is marked as state S_9.

According to definition 1, user behavior navigation model is formalized as follows:
$S = \{S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9\}$, initial state $=\{S_0\}$, $R = \{(S_0, S_1),(S_0, S_2), (S_0, S_3), (S_0, S_9), (S_1,S_4), (S_1, S_5), (S_1, S_7), (S_1, S_8), (S_1, S_9), (S_2, S_5), (S_2,S_6), (S_2, S_8), (S_2, S_9), (S_3, S_5), (S_3, S_6), (S_3, S_8), (S_3,S_9), (S_4, S_7), (S_4, S_8), (S_4, S_9), (S_5, S_8), (S_5, S_9), (S_6,S_8), (S_6, S_9), (S_7, S_8), (S_7, S_9)\}$, $L(S_0) = \{ S_0= true\}$, $L(S_1) = \{ S_1= true\}$, $L(S_2) = \{ S_2 = true\}$, $L(S_3) = \{ S_3 = true\}$, $L(S_4) = \{ S_4 = true\}$, $L(S_5) = \{ S_5 = true\}$, $L(S_6) = \{ S_6 = true\}$, $L(S_7) = \{ S_7 = true\}$, $L(S_8) = \{ S_8 = true\}$, $L(S_9) = \{ S_9= true\}$.
After determining the functional behavior of the service recommendation system, the possible transition between the different states depends on user context information, user preferences, and user's previous state, in combination with the invalid recommendation, the transition probability of navigation behavior can be described by the migration probability matrix. As shown in Table II, there is a transition probability matrix from one state to another state and the corresponding cell indicates the probability of successful service recommendation.
We can also use Markov chain to describe the user behavior transition probability. As shown in Figure 5, the circle represents a status of the user, a curve line with an arrow represents user' status transition, the

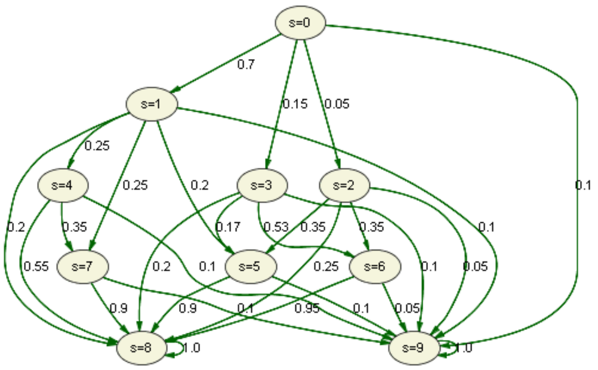

Fig. 6. Markov chain usage model

arrow represents the conversion direction. The number on a straight line represents the probability value from one state transition to another.
We adopt a probabilistic model, discrete time Markov chain (DTMC), to describe the behavior of the recommended Web services, and validate the specified attributes in the form of a temporal logic (PCTL) and evaluated by probabilistic model checker PRISM. Depending on the different properties, the probability value r may be the probability of the start service Si reaching successfully the target service St while satisfying the property, it can also be a probability of from the initial state reaching successfully to any state.
The qualitative validation of the DTMC is judged by probability reachability, while the quantitative properties verification focuses on the state coverage using PCTL. From state S_0 to end state S_8 or S_9, whether the reachability can be judged by the following property

$$P_{=?}(F \text{ state} = 8 \text{ or } 9) \tag{8}$$

To verify the recommended Web service reliability, we translate the DTMC model into language PRISM. Table III shows PRISM code’s variables and guarded commands. The variable is defined as enumeration type with initial value, such as “state:[S0, S1, S2, S3, S4, S5, S6, S7, S8, S9] init 0”. And The protection command specifies the transition probability. For example, in line 5, when service state S_0 is visited, the transition will evolve to service state S_1, service state S_2 ,service state S_3 and service state S_9 with probability of 70%, 5% , 15% and 10% respectively.

Table 2. The migration probability matrix

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_0	-	0.7	0.05	0.15	-	-	-	-	-	0.1
S_1	-	-	-	-	0.25	0.2	-	0.25	0.2	0.1
S_2	-	-	-	-	-	0.35	0.35	-	0.25	0.05
S_3	-	-	-	-	-	0.17	0.53	-	0.2	0.1
S_4	-	-	-	-	-	-	-	0.35	0.55	0.1
S_5	-	-	-	-	-	-	-	-	0.9	0.1
S_6	-	-	-	-	-	-	-	-	0.95	0.05
S_7	-	-	-	-	-	-	-	-	0.9	0.1
S_8	-	-	-	-	-	-	-	-	-	-

Table 3. Prism code

No	Statement
[1]	dtmc
[2]	module RWS
[3]	//local variables
[4]	s:[0.. 9] init 0;
[5]	[ ] s = 0 -> 0.7: (s' = 1)+0.05 : (s' = 2)+0.15 : (s' = 3)+0.1: (s' = 9);
[6]	[ ] s =1->0.25: (s' = 4)+0.2 : (s' =5)+0.25: (s' = 7)+0.2 : (s' = 8)+0.1: (s' = 9);
[7]	[ ] s =2 ->0.35 : (s' = 5)+0.35: (s' = 6)+0.25: (s' = 8)+0.05: (s' = 9);
[8]	[ ] s = 3 -> 0.17 : (s' = 5) + 0.53 : (s' = 6)+0.2: (s' = 8)+0.1: (s' =9);
[9]	[ ] s = 4 -> 0.35 : (s' = 7)+0.55 : (s' = 8)+0.1: (s' = 9);
[10]	[ ] s = 5 -> 0.9 : (s' = 8)+0.1 : (s' =9);
[11]	[ ] s = 6 -> 0.95 : (s' =8)+0.05 : (s' = 9);
[12]	[ ] s = 7 -> 0.9 : (s' =8)+0.1 : (s' = 9);
[13]	endmodule

Through the automatic verification of the model checker PRISM, the data of Table 4 are obtained and it shows the total probability value from the specified state to the desired turntable in this case.

(i) Formula $P_{=?}$ (F state = 1) calculates the probability of starting from the initial state to the specified state of 1, and indicates the reachability probability

(ii) The probability value is 0 or 1 indicates that it is a universal quantification.

(iii) State S8 and S9 are the terminal state of this example. $P_{=?}$(F state = 8) inditcates the probability of reaching the terminal state from each inner state. $P_{=?}$(F state = 9) indicates the probability of reaching failure state. The remaining items in the table indicates the reachability probability of the other internal states in this example.

What we should do is adding an invocation response time constraint condition in the original property. In this illustration, we focus on the property $P_{=?}$ (F state = 8) starting with service state S_0 in order to make the linear regression equation clear to evaluate the reliability of reccomended services.The new property as as followings $P_{=?}$ [F (s=8) & x<=5] where variable x represents the service response time. Table IV shows the probability of reaching the end state with response time condition constraint. According to the formula (3) and the data of table IV, we can compute the two constants values of a and b in the regression linear equation.

Table 4. Probabilistic model checking results

Property Formula	Start State	Probability
	State =0	1
	State =1	0
	State =2	0
	State =3	0
$P_{=?}$(F state = 0)	State =4	0
	State =5	0
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.7
	State =1	1
	State =2	0
	State =3	0
$P_{=?}$(F state = 1)	State =4	0
	State =5	0
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.05
	State =1	0
	State =2	1
	State =3	0
$P_{=?}$(F state =2)	State =4	0
	State =5	0
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.15
	State =1	0
	State =2	0
	State =3	1
$P_{=?}$(F state = 3)	State =4	0
	State =5	0
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.175
	State =1	0.25
	State =2	0
	State =3	0
$P_{=?}$(F state = 4)	State =4	1
	State =5	0
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.183
	State =1	0.2
	State =2	0.35
	State =3	0.17
$P_{=?}$(F state = 5)	State =4	0
	State =5	1
	State =6	0
	State =7	0
	State =8	0
	State =9	0
	State =0	0.097
$P_{=?}$(F state = 6)	State =1	0
	State =2	0.35

Table 4 (Continued)

Property Formula	Start State	Probability
	State =3	0.53
	State =4	0
	State =5	0
	State =6	1
	State =7	0
	State =8	0
	State =9	0
$P_{=?}$(F state = 7)	State =0	0.236
	State =1	0.338
	State =2	0
	State =3	0
	State =4	0.35
	State =5	0
	State =6	0
	State =7	1
	State =8	0
	State =9	0
$P_{=?}$(F state = 8)	State =0	0.748
	State =1	0.821
	State =2	0.898
	State =3	0.857
	State =4	0.865
	State =5	0.899
	State =6	0.95
	State =7	0.899
	State =8	1
	State =9	0
$P_{=?}$(F state = 9)	State =0	0.252
	State =1	0.179
	State =2	0.105
	State =3	0.144
	State =4	0.135
	State =5	0.1
	State =6	0.05
	State =7	0.1
	State =8	0
	State =9	1

In this illustration, we focus on the property $P_{=?}$ (F state = 8) starting with service state S0 in order to make the linear regression equation clear to evaluate the reliability of reccomended services.The new property as as followings $P_{=?}$ [F (s=8) & x<=5] where variable x represents the service response time. Table IV shows the probability of reaching the end state with response time condition constraint. According to the formula (3) and the data of table IV, we can compute the two constants values of a and b in the regression linear equation.

Table 5. Data of linear equations

ID	Time(t)	Probability(y)	t^2	ty
1	1	0.821	1	0.821
2	2	0.804	4	0.804
3	3	0.785	9	2.355
4	4	0.757	16	3.028
5	5	0.721	25	3.605
Sum	15	3.888	55	10.613

In Table 5, the meaning of each column is as follows:

- The item "ID" in table IV indicates the number of times that data is obtained. In our experimentation, we get user context, and dynamically build the model and calculate every 10 seconds while mobile user is moving.
- The item “time” indicates constraint - response time.
- The item “Probility” indicates Indicates the probability value after adding this constraint.

According to linear regression equation calculation method introduced in section II, we can calculate:

$$b = \frac{n\sum ty - \sum t \sum y}{n\sum t^2 - (\sum t)^2} = \frac{5*10.613 - 15*3.888}{5*55 - 15^2} = -0.1051$$

$$a = \frac{\sum y}{n} - b\frac{\sum t}{n} = \frac{3.888}{5} - b*\frac{15}{5} = 1.0929$$

Since the coefficients a and b are determined, we can conclude that the linear regression equation is shown as below:

$$y = 1.0929 - 0.1051*t$$

We experimented on a computer with 2.4 GHz CPU and 4 GB RAM and Windows 7 Operating System. Table IV shows the probability model verify results in such a simulation environment. The result of the property check indicates the reliability probability of the recommended service to the user when the user passes through each location.

The linear regression equation is solved by the property $P_{=?}$[s=B->f(End)] as an example. Calculate the values of the linear regression equations a and b based on the resulting data. According to the above formula, the linear equation y = 1.0929 -0.1051*t is obtained, which indicates that the service recommendation for the visitor from the state S0 transit to the state End. The probability of the service successful recommendation is 0.8827 with response time<=2 seconds.

With the increase of the constraint time, the probability of successful service recommendation decreases at the rate of1.09, indicating that the longer the response time, the greater the probability of recommended service failure. The context collection phase collects the user context information for the first period which is divided into ten fragments. The blue line is the trend of the value of service reliability. It can be seen that as the time constraint value t increases, the likelihood of a

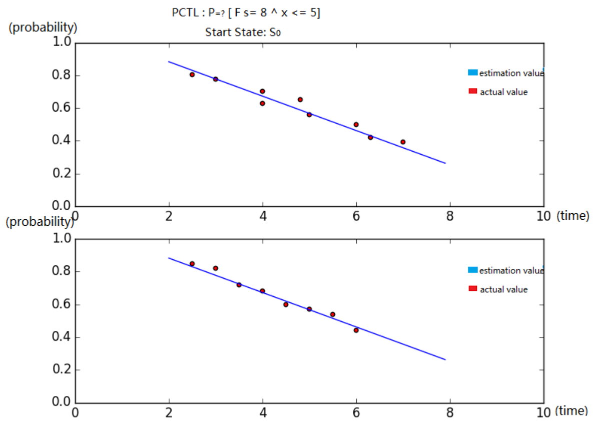

Fig. 7. Comparison of actual value of service reliability and estimation value

recommended service failure is greater. We can draw a linear regression equation based on the user's choice of historical data for the recommended service.

Figure 7 illustrates a terminal state reachable probability value trend for a mobile user choosing different Web with same function. The closer the actual value is to the evaluation value, the more matching up the recommended service and the user needs under the same response time constraint.

We can save the Web service composition as a preferred history record which actual value is close to the value of the evaluation. The next time, when user needs services, the system detects that user context and the history record are similar, Web service comprision in historic records will be a priority recommendation.

6. Conclusions And Future Work

In business process events, more research is about the reachability of the recommended service, that is, a path from the initial state to the final state is reachable. In the intelligent service recommendation process, each state is random. The state of the conversion is also random. It is only necessary to qualitative and quantitative analyze the services recommended for the current state and its successor reliability by adopting parameter estimation method, making the recommendation more accurate.

In this paper, we concentrated on studying the evaluation of the reliability of the recommended service. Our approach complete the mapping of the formatted context information to the desired service state and construct a DTMC model for performing probabilistic model checking. Then, the PCTL formulae extracted by state converge were used as the quantitative property of recommended Web service. After executing formal verification in model checker PRISM, we introduced a statistics method to show the linear equation for revealing the relationship between the reliability and the invocation duration time. Finally, a case study was discussed, and experiment results shown that our approach had a good result.

We believe that the application of the reliability evaluation of recommended services has broad prospects. It can be used for service function monitoring, service performance testing, service adaptive adjustment and so on. For future work, We also plan to systematize non-functional verification in addition to response time, also verify the availability, cost and other properties. At the same time, the correspondence between user history selection and context information will be studied.

Acknowledgements

This paper is supported by National Natural Science Foundation of China (NSFC) under Grant No. 61572306 and No. 61170044.

References

1. Zheng Z, Ma H, Lyu M R, et al. Qos-aware Web service recommendation by collaborative filtering[J]. IEEE Transactions on services computing, 2011, 4(2): 140-152.
2. Chen G, Kotz D. A survey of context-aware mobile computing research[R]. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
3. Cordeiro J, Antunes B, Gomes P. Context-based recommendation to support problem solving in software development[C]//Recommendation Systems for Software Engineering (RSSE), 2012 Third International Workshop on. IEEE, 2012: 85-89.
4. Bentahar J, Yahyaoui H, Kova M, et al. Symbolic model checking composite Web services using operational and control behaviors[J]. Expert Systems with Applications, 2013, 40(2): 508-522.
5. Dıaz G, Cambronero M E, Pardo J J, et al. Model checking techniques applied to the design of Web services[J]. CLEI Electron. J, 2007, 10(2).
6. Sheng Q Z, Maamar Z, Yao L, et al. Behavior modeling and automated verification of Web services[J]. Information Sciences, 2014, 258: 416-433.
7. Su G, Rosenblum D S, Tamburrelli G. Reliability of run-time quality-of-service evaluation using parametric model checking[C]//Proceedings of the 38th International Conference on Software Engineering. ACM, 2016: 73-84.

8. Babin G, Ameur Y A, Pantel M. Formal verification of runtime compensation of Web service compositions: A refinement and proof based proposal with Event-B[C]//Services Computing (SCC), 2015 IEEE International Conference on. IEEE, 2015: 98-105.
9. Moreno G A, Cámara J, Garlan D, et al. Proactive self-adaptation under uncertainty: a probabilistic model checking approach[C]//Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. ACM, 2015: 1-12.
10. Belli F, Endo A T, Linschulte M, et al. A holistic approach to model - based testing of Web service compositions[J]. Software: Practice and Experience, 2014, 44(2): 201-234.
11. Chen M, Tan T H, Sun J, et al. Verification of functional and non-functional requirements of web service composition[C]//International Conference on Formal Engineering Methods. Springer, Berlin, Heidelberg, 2013: 313-328.
12. Mobasher B, Jin X, Zhou Y. Semantically enhanced collaborative filtering on the web[M]//Web Mining: From Web to Semantic Web. Springer Berlin Heidelberg, 2004: 57-76.
13. Jiang Y, Liu J, Tang M, et al. An effective web service recommendation method based on personalized collaborative filtering[C]//Web Services (ICWS), 2011 IEEE International Conference on. IEEE, 2011: 211-218.
14. Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.
15. Park M H, Hong J H, Cho S B. Location-based recommendation system using bayesian user's preference model in mobile devices[C]//International Conference on Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg, 2007: 1130-1139.
16. Chou P H, Li P H, Chen K K, et al. Integrating Web mining and neural network for personalized e-commerce automatic service[J]. Expert Systems with Applications, 2010, 37(4): 2898-2910.
17. Popescul A, Pennock D M, Lawrence S. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments[C]//Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 2001: 437-444.
18. Dey A K. Understanding and using context[J]. Personal and ubiquitous computing, 2001, 5(1): 4-7.
19. Baldauf M, Dustdar S, Rosenberg F. A survey on context-aware systems[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2007,2(4):263-277.
20. Tang M, Jiang Y, Liu J, et al. Location-aware collaborative filtering for QoS-based service recommendation[C]//Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012: 202-209.
21. Gavalas D, Kenteris M. A webWeb-based pervasive recommendation system for mobile tourist guides[J]. Personal and Ubiquitous Computing, 2011, 15(7): 759-770.
22. Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[M]//Applications of Data Mining to Electronic Commerce. Springer US, 2001: 115-153.
23. Wen W, Miao H. Context-Based Service Recommendation System Using Probability Model in Mobile Devices[C]//Enterprise Systems (ES), 2016 4th International Conference on. IEEE, 2016: 178-182.
24. Xia H, Yoshida T. Web service recommendation with ontology-based similarity measure[C]//Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on. IEEE, 2007: 412-412.
25. Wang ML, Chen BH, Peng X, Huang G, Zhao WY. Personal service publishing and composition in mobile computing environment. Ruan Jian Xue Bao/Journal of Software, 2015,26(4):802−818 (in Chinese). http://www.jos.org.cn/1000-9825/4754.htm
26. Gao H, Miao H, Zeng H. Predictive web service monitoring using probabilistic model checking[J]. Appl. Math, 2013, 7(1L): 139-148.
27. R.L. Huang and Y.H Guan. Data Statistics Analysis: SPSS Theory and Applications. The Higher Education Press, Beijing (2010), (in Chinese)
28. Gavalas D, Konstantopoulos C, Mastakas K, et al. Mobile recommender systems in tourism[J]. Journal of network and computer applications, 2014, 39: 319-333.
29. Zheng V W, Cao B, Zheng Y, et al. Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach[C]//AAAI. 2010, 10: 236-241.
30. Zhou B, Dastjerdi A V, Calheiros R N, et al. A context sensitive offloading scheme for mobile cloud computing service[C]//Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on. IEEE, 2015: 869-876.
31. He S, Chan S H G. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons[J]. IEEE Communications Surveys & Tutorials, 2016, 18(1): 466-490.
32. Yang Z, Wu C, Zhou Z, et al. Mobility increases localizability: A survey on wireless indoor localization using inertial sensors[J]. ACM Computing Surveys (Csur), 2015, 47(3): 54.
33. H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," Formal Aspects of Computing, vol. 6, no. 5, pp. 512–535, 1994.