Service Composition and User Modeling for Personalized Recommendation in Cloud Computing

Guobing Zou
School of Computer Engineering and Science
Shanghai University
Shanghai 200444, China
Email: gbzou@shu.edu.cn

Yanglan Gan
School of Computer Science and Technology
Donghua University
Shanghai 201620, China
Email: ylgan@dhu.edu.cn

Jianxing Zheng
School of Computer Engineering and Science
Shanghai University
Shanghai 200444, China
Email: jxzheng185@163.com

Bofeng Zhang
School of Computer Engineering and Science
Shanghai University
Shanghai 200444, China
Email: bfzhang@shu.edu.cn

***Abstract*—In recent years, cloud computing is gradually evolving as a popular computing paradigm, which offers a uniform platform for service providers to publish their applications as cloud services. In many cases, however, single cloud service cannot satisfy a service request due to its simple functionality. Furthermore, current service composition systems have seldom taken into account user interests for personalized recommendation. In this paper, we propose a novel framework for personalized service recommendation in cloud computing platform by Web service composition and user modeling. The proposed framework first models cloud services together with a service request as a Web service composition problem, called cloud service recommendation (CSR) planning problem. It is fed into our self-developed service planner to compose a cloud service with complex business workflow. Second, our framework also applies user modeling for checking whether the generated composite cloud service can be matched with the interests of service consumer. To validate the feasibility of CSR framework, we have designed and implemented two prototype systems, QoS-aware service composition system and service platform based on user model.**

I. Introduction

Web services are modular, self-descriptive, loosely coupled, and accessible distributed software components. After wrapping the functionality of an application and providing accessible interfaces, they can be published over the Internet, discovered by software agents and composed as new value-added and cross-organizational distributed applications [1]. From the perspective of Service-Oriented Architecture (SOA), Web services have become one of the most important components to dynamically create distributed applications as requested. However, the underlining technique for SOA design and implementation is referred to as Web service composition (WSC), where no single service can be satisfied for a service request. That is, WSC aims at composing a chain of connected services from a Web service repository to create a new and value-added composite service. WSC can be applied in a variety of application scenarios, such as business workflow management, electronic commerce, and enterprise application integration.

Cloud computing is gradually evolving as a popular computing paradigm. From the view of both application and infrastructure, [2] defined cloud computing as both the applications delivered as services over the Internet and the hardware and system softwares in the data centers. Currently, popular cloud platforms include Google File System, IBM Blue Cloud, Amazon Elastic Computing Cloud (EC2) and Tsinghua transparent computing system [3]. Cloud service providers publish their cloud applications in the level of Software as a Service (SaaS). However, in most cases these applications are provided for cloud consumers in a cloud platform with only single functionalities. As a result, single cloud service cannot satisfy end consumers if they need to find a composite service with several connected cloud services by multiple invocation relationships, such as sequential, parallel, and conditional ones. Therefore, how to efficiently wrap cloud resources as standard services and dynamically compose a set of these cloud services as a whole to serve for cloud consumers has become an open research problem.

Furthermore, user modeling is becoming a powerful tool to describe user interests by formal and conceptual languages. There are two popular ways for the construction of user model: content-based user modeling and semantic-based user modeling. They have been applied in real-world applications for personalized recommendation. However, user's interest features are always complex and difficult to capture. Thus, how to construct user modeling and apply it to Web service composition for cloud service recommendation is a research issue to be resolved in cloud computing platform.

To solve above research challenges, this paper proposes a novel framework for cloud composite service recommendation. To the best of our knowledge, it is the first time to propose a framework that integrates service composition and user modeling into cloud computing platform for personalized service recommendation. It depends on our previous research works in dynamic service composition [1] and user modeling [4]. Our approach firstly translates cloud services with cloud QoS values and service interest features to a cloud service planning domain with formal planning description languages. Then, when a service consumer submits a cloud service request, our approach formulates a Web service composition problem, called cloud service recommendation (CSR) planning problem. As a consequence, depending on the CSR planning problem, we employ an off-the-shelf highly efficient automated planner (e.g. Metric-FF [5]) or our self-developed planner (i.e., CSTE planner [1], [6]) to generate a cloud service, which is a composite service with a chain of connected services. Furthermore, to check the satisfiability of generated composite service, we apply user modeling techniques to matching the cloud service interests. By doing so, our approach provides personalized cloud service recommendation.

The proposed cloud service recommendation framework is supported by our two implemented prototype systems in Java. The prototype system for dynamic composition of Web services takes composition request with multiple QoS constraints and user preferences as inputs, and it generates a composite service. Service platform based on user model constructs user model from the perspective of content interest degree and semantic interest degree, which can provide user interests for personalized recommendation. In our future experimental evaluation, these two prototype systems will be seamlessly integrated into a self-constructed cloud computing environment for cloud service recommendation.

The rest of this paper is organized as follows. Section II reviews related work. Section III formulates our research problem. Section IV describes our approach for cloud service recommendation. Section V presents our two implemented prototype systems on service composition and user modeling, respectively. Finally, Section VI concludes our paper and discusses the future work.

II. Related Work

A. Dynamic Composition of Web Services

According to the applied theories and techniques [7], WSC methods can be classified as workflow-based, AI planning based, graph theory based, and program synthesis based methods. However, we mainly investigate planning-based dynamic composition of Web services, since we apply automated planning as the fundamental techniques for cloud service recommendation.

Many research efforts using AI planning techniques have been reported in recent years. Web service planner (WSPR) [8] presented an AI planning based algorithm to implement automatic composition of Web services. To find a feasible composition solution, it goes through two phases including forward search and regression search. During its search for a composition solution, a heuristic function is used to choose a service with the biggest contribution to match a subgoal. The authors in [9] proposed a service composition algorithm by planning graph model. The process of finding a composition solution is the construction of a planning graph. Although it is a complete algorithm, when a new planning graph level is expanded by a set of applicable services, it just selects a subset of services in order, until they cover all of the output parameters of the candidate services. A generic framework for service composition was presented in [10], where a service repository described by OWL-S is translated to an AI planning domain in PDDL.

Besides the above planning based WSC methods, Automated planners have been applied to address a WSC problem. SHOP2 is a Hierarchical Task Network (HTN) planning system, which has been exploited for automatic Web service composition in [11]. All of the available services are first translated to a SHOP2 domain, and then SHOP2 planner recursively divides a composition task into many subtasks, until every subtask can be executed by a single service. Hoffmann et al [12] presented a planning-based method to formalize a special case WSC problem called “Strictly Forward Effects”. It takes integrity constraints as background theory specified by ontology to describe domain constraints.

B. User Modeling

User modeling technique is used to describe user interests by formal and conceptual languages. In the user model construction, how to convert the raw documents into user’s interest subjects is significant. A variety of construction methods of user model have proposed, such as content-based user modeling and semantic-based user modeling.

Content-based user model focuses on document content analysis to classify the categorization of user’s browsing historical records for deriving the hot interest and meaningful subjects. Considering the blogosphere unique characteristics, Agarwal et al. [13] proposed collective wisdom to cluster blogs to serve the needs of users. Mei et al. [14] presented a mechanism to explore the distributions and evolution models across time and space from the contents. Lee [15] applied a density-based online clustering method for mining micro-blogging text streams to obtain temporal and geospatial features of real-world events, and proposed comprehensive spatio-temporal viewpoint to satisfy their information needs. Considering the characteristic of short text messages, Esparza [16] described users and products from the terms used in abbreviated and highly personalized commentary and studied how Twitter-like short-form messages could be used as a source of indexing and retrieval information for product recommendation.

Semantic-based user model focuses on investigating the semantic links to obtain user’s interests for generating personal user model. Taking into account semantic information and word order information, Li [17] proposed an algorithm to model human common sense knowledge representation and knowledge discovery through computing the similarity between two sentences. Based on WordNet and multiple ontologies, Ray et al. [18] proposed blog content theme expansion approach and a ranking algorithm for blog content in order to recommend conceptually related expanded themes of blog content. Another ontological approach over user model in recommendation systems is introduced in [19]. The authors proposed two systems, Quickstep and Foxtrot, which learn user model by monitoring the user’s behaviors and gathering relevance feedback information from them in the process of recommending on-line academic research papers.

III. Problem Formulation

This section focuses on the formulation of cloud service recommendation problem using the techniques of planning-based Web service composition and user modeling. We first formulate cloud service and user modeling, then define the problem of cloud service recommendation.

Definition 3.1: ***(Cloud Service)***. A service s is defined as a 4-tuple (I, O, Q, U), where $I = \{I^1, I^2, \cdots\}$ is a set of input parameters, $O = \{O^1, O^2, \cdots\}$ is a set of output parameters, $Q = \{Q^1, Q^2, \cdots\}$ is a set of QoS values that represent the non-functional values of s, and $U = \{U^1, U^2, \cdots\}$ is a set of service domain features that represent the interests of s provided for cloud service consumers, where each U^i is a concept from a domain ontology. We use $s.I$, $s.O$, $s.Q$, and $s.U$ to denote I, O, Q, and U of s, respectively.

Each cloud service plays a role that can perform a specified task. A cloud service repository is a set of disjoint services. It is defined as follows.

Definition 3.2: ***(Cloud Service Repository)***. A cloud service repository $S = \{s_1, s_2, \cdots\}$ is a set of cloud services. Where, each $s \in S$ is a service.

Given a cloud service $s = (I, O, Q, U)$, its multiple QoS values Q correspond to a set of cloud QoS criteria as below.

Definition 3.3: ***(Cloud QoS Criteria Set)***. Cloud QoS criteria q represents a dimension of non-functional criteria in a cloud service s. QoS criteria set, $qos = \{q^1, q^2, \cdots, q^n\}$, are a set of QoS properties. For $\forall q^i \in qos(1 \leq i \leq n)$, it corresponds to a QoS value $Q^i \in s.Q$ in a cloud service s.

Cloud QoS criteria can be divided into two categories: positive and negative. Positive cloud QoS criteria denote better quality with higher values, while negative ones correspond to lower quality with higher values.

Based on widely used QoS criteria [20], we apply a QoS criteria set $qos = \{q^1, q^2, \cdots, q^n\}$ to model and specify the QoS values of each cloud service s. More specifically, we employ *qos*=\{*cloud time, cloud price, cloud availability, execution success rate, reputation*\} as references of QoS criteria for cloud service providers to offer QoS values for a cloud service. As mentioned above from the classification of cloud QoS criteria, *cloud time* and *cloud price* are clarified as negative QoS properties, while *cloud availability*, *execution success rate*, and *reputation* are positive ones.

Definition 3.4: ***(Cloud Consumer Preferences)***. Given a set of cloud QoS criteria $qos = \{q^1, q^2, \cdots, q^n\}$, cloud consumer preferences, denoted as $W = \{w_1, w_2, \cdots, w_n\}$, are a set of corresponding QoS weights. For each $q^i (1 \leq i \leq n)$, a cloud service consumer can assign a QoS weight $w_i (1 \leq i \leq n)$ to express the preference on the criterion, while all the weights must satisfy $\sum_{i=1}^{n} w_i = 1$ and $0 \leq w_i \leq 1$.

In addition to cloud consumer preferences attached on a set of given cloud QoS criteria $qos = \{q^1, q^2, \cdots, q^n\}$, a cloud service consumer always submits multiple QoS constraints on each criterion to enhance the satisfiability of non-functional features for desirable cloud service.

Definition 3.5: ***(Cloud QoS Constraints)***. Given a set of cloud QoS criteria $qos = \{q^1, q^2, \cdots, q^n\}$, $C = \{c^1, c^2, \cdots, c^n\}$ is denoted as a set of cloud QoS constraints, where each $c^i \in C$ is a constraint on the corresponding cloud QoS criterion $q^i \in qos(1 \leq i \leq n)$.

Depending on the cloud QoS classifications, for a positive cloud QoS criterion $q^i \in qos(1 \leq i \leq n)$, we set $c^i = (q^i, \geq, v^i)$ as a cloud QoS constraint, where v^i is the lower bound of q^i. In the same way, for a negative cloud QoS criterion $q^j \in qos(1 \leq j \leq n)$, $c^j = (q^j, \leq, v^j)$ is denoted as a cloud QoS constraint, where v^j is the upper value of q^j.

Definition 3.6: ***(Cloud Service Request)***. A cloud service request is a 4-tuple, denoted as $r = (I_r, O_r, C, W)$, where $I_r = \{I_r^1, I_r^2, \cdots\}$ is a set of input parameters provided by a cloud service consumer as initial conditions, $O_r = \{O_r^1, O_r^2, \cdots\}$ is a set of output parameters desired by the cloud service consumer. C and W are cloud QoS constraints and cloud consumer preferences, respectively.

Definition 3.7: ***(User Model)***. Given a domain ontology DO, a user model for a cloud service consumer, denoted as $U_c = \{(U_c^1, Cid_1, Sid_1), (U_c^2, Cid_2, Sid_2), \cdots\}$, where $U_c^i \in DO$ is an ontology concept that specifies one facet of user interests, Cid_i represents the content interest degree on U_c^i, and Sid_i represents the semantic interest degree on U_c^i.

Given a set of available disjoint cloud services, a cloud service request, and the user model of a cloud service consumer, we can formally define the problem of cloud service recommendation (CSR) in the following.

Definition 3.8: ***(Cloud Service Recommendation, CSR)***. A cloud service recommendation problem is defined as a 6-tuple, denoted as CSR=(S, C, W, U_c, I_r, O_r), where (1) $S = \{s_1, s_2, \cdots\}$ is a cloud service repository, (2) $C = \{c^1, c^2, \cdots, c^n\}$ is a set of cloud QoS constraints on the specified cloud QoS criteria, (3) $W = \{w_1, w_2, \cdots, w_n\}$ refers to corresponding cloud consumer preferences on the cloud QoS criteria, (4) $U_c = \{(U_c^1, Cid_1, Sid_1), (U_c^2, Cid_2, Sid_2), \cdots\}$ is the set of user interests and associated degrees extracted from domain ontology and real-world applications, (5) $I_r = \{I_r^1, I_r^2, \cdots\}$ is an initially functional condition with a set of input parameters, and (6) $O_r = \{O_r^1, O_r^2, \cdots\}$ is a goal specification with a set of desirable output parameters.

The cloud service recommendation problem defines a service composition problem with QoS optimization and user model requirement. The solution to a cloud service recommendation $CSR = (S, C, W, U_c, I_r, O_r)$ is a composite service with a chain of connected cloud services. The generated composite service can match the cloud service request $r = (I_r, O_r, C, W)$ from the view of functionality I_r and O_r, cloud QoS constraints C and QoS optimization of cloud service using a weighted sum of preferences W, while the corresponding user model U_c of the cloud service consumer must be satisfied by a combination of interests of cloud services included in the composite service.

IV. Cloud Service Recommendation

Cloud service recommendation can be divided into two steps. The first one is to dynamically compose a chain of connected cloud services by using planning-based Web service composition approach based on our previous work [1], which includes planning domain translation of cloud services and planning of generating a composite cloud service. In the second step, user models of cloud service consumers are constructed and the semantic matching between user model and a combination of interest features of cloud services is checked for the satisfiability of the composite cloud service for personalized service recommendation. The framework of cloud service recommendation is illustrated in Figure 1.

A. Framework of Cloud Service Recommendation

To validate the feasibility of our proposed approach using planning-based Web service composition and user modeling, we propose an architecture of cloud service recommendation. The system architecture is outlined in Figure 1. It consists of several modules, including a CSR graphical user interface (GUI), user modeling, user model checker, cloud service development, CSR planning translator, cloud service planner, and cloud service infrastructure.

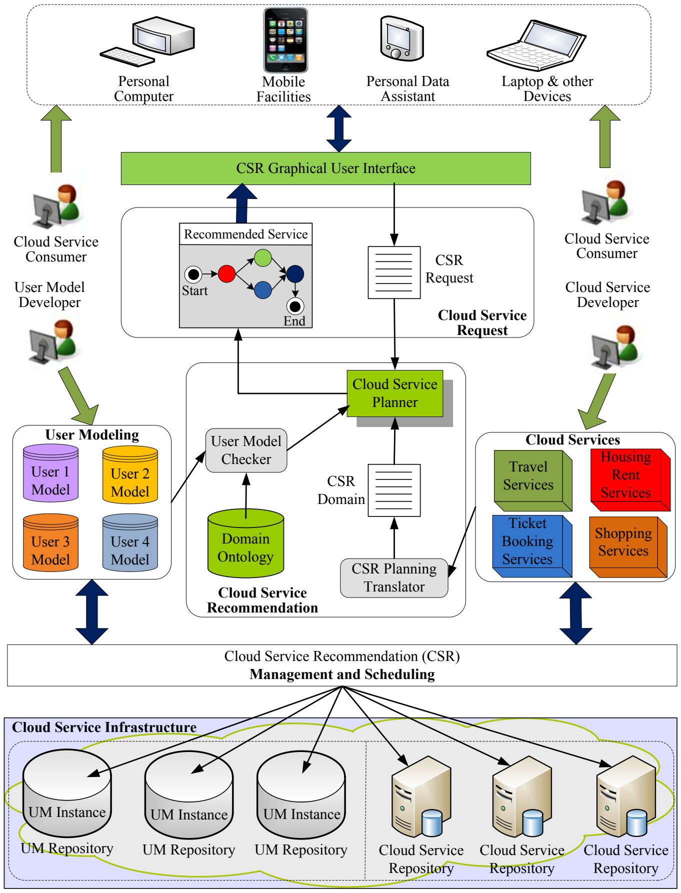

Figure 1. The framework of personalized cloud service recommendation.

The process of cloud service recommendation based on planning-based Web service composition and user modeling is comprised of four steps. First, cloud service developers gather the cloud services from CSR infrastructure by CSR management and scheduling. These cloud services are translated to a cloud service recommendation planning domain by CSR planning translator. Second, cloud service consumers provide their interests by selecting semantic concepts from domain ontology, so that user model developers organize and store user models into the system. Third, a cloud service consumer submits a request by GUI, which is translated to a CSR request with formal expression. Finally, based on the generated cloud service request, user model and CSR planning domain, the CSR system invokes a cloud service planner and user model checker to generate a composite cloud service that is composed by a chain of connected cloud services and returned to the cloud service consumer.

B. Planning Transition of CSR Problem

Based on our previous work [1], the cloud service planning problem and cloud service action are defined as below.

Definition 4.1: **(Cloud Service Planning Problem)**. A cloud service planning problem is defined as (A, L, V, s_0, g), where A is a set of cloud service actions, L is a set of logical facts, V is a set of numeric variables, s_0 is the initial state, and g is a goal specification.

A cloud service planning problem describes a planning problem with logical reasoning, cloud response time temporal constraints planning, and cloud QoS optimization.

Definition 4.2: **(Cloud Service Action)**. A cloud service action a is defined by a tuple (pre, eff, μ, ρ), where

- $pre(a)$ is the action precondition that consists of a set of numeric constraints based on numeric variables V and a set of logical facts, each of which is an atomic proposition $l \in L$.
- $eff(a)$ is the action effect that consists of a set of numeric effects based on V and a set of logical facts from L.
- $\mu(a)$ is the action duration of cloud service.
- $\rho(a)$ is the action cost of cloud service.

A planning solution to a cloud service planning problem (A, L, V, s_0, g), is a composite cloud service, which is a dependency graph $G^* = (V, E)$. It transforms the initial state s_0 to a goal state g^*, such that all the logical facts in g are subsumed in g^*, i.e., $g \subseteq g^*$. Moreover, an optimal solution to a cloud service planning problem (A, L, V, s_0, g) minimizes the cost of all cloud service actions in G^*, denoted as $\min \sum_{a_i \in G^*} \rho(a_i)$.

Given a cloud service recommendation problem $CSR = (S, C, W, r_{in}, r_{out}, U)$, we first translate it to a cloud service planning problem (A, L, V, s_0, g) without the consideration of user model U. The process of planning translation of CSR problem is as follows.

(1) For each cloud service $s \in S$, we use I, O, Q from the service $s = (I, O, Q, U)$ and transform it to a cloud service action $a = (pre, eff, \mu, \rho)$. Precondition $pre(a)$ and effects $eff(a)$ are translated by $s.I$ and $s.O$ from logical parts, while cloud QoS constraints C and predefined global QoS variables V on QoS criteria are applied to check the satisfiability of global cloud QoS constraints in $pre(a)$ and update the cloud QoS effects in $eff(a)$. The duration $\mu(a)$ of a cloud service action a is transformed from the QoS value of cloud time in $s.Q$, and the cost $\rho(a)$ of a cloud service action a is set by a weighted sum of normalized QoS values of $s.Q$. (2) The logical facts of cloud service planning problem include all precondition and effect facts by input and output parameters from every cloud service $s \in S$. That is, for each $I^i \in s.I$ or $O^i \in s.O$, we add a logical fact to L by first-order predicates. (3) Variables V in cloud service planning problem are divided into four groups to represent objective QoS value function, global QoS constraints of composite service, QoS effects of cloud service, and service consumer preferences. (4) Initial state s_0 is set by the initial logical facts from r_{in} and initial cloud QoS values of the four kinds of variables from V and W. (5) Goal specification g is set by goal parameters r_{out}.

C. Cloud Service Planning Problem Solving

Given a cloud service planning instance (A, L, V, s_0, g) transformed from a cloud service recommendation problem $CSR = (S, C, W, r_{in}, r_{out}, U)$, we solve it by our previous developed service composition planner called CSTE planner [6], which translates our problem to an optimization problem with Satisfiability (SAT) constraints and global QoS constraints. Furthermore, it is solved by our planner that generates a composite service as a solution to our cloud service recommendation problem without checking the satisfiability of user model. It is a dependency graph which describes the correct invocation and execution order of actions.

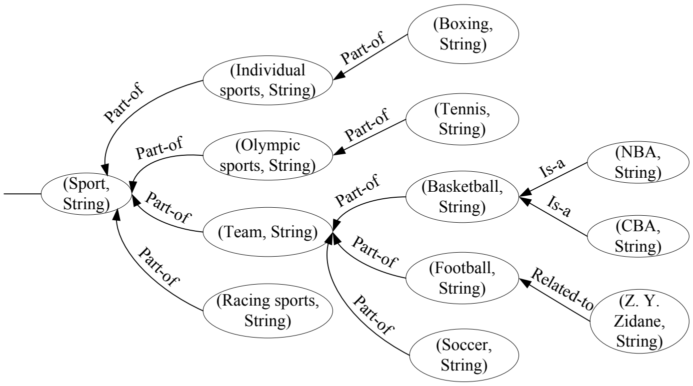

Figure 2. A sample structure for "sport" in a domain ontology.

D. User Interests Modeling

The construction of a personalized user model based on our previous work [4], which extracts the interest degree derivation of subject. The interest degree consists of two components: content interest degree and semantic interest degree.

1) Content Interest Degree of Subject: Documents recorded usually contain different interest subjects. In our work, text preprocessing of stop word removal and word stemming is processed to deal with the text. It is finally represented as $d = \{(t_1, w_1), (t_2, w_2), \cdots, (t_p, w_p)\}$ formed by a set of term weighting pairs as follows.

$$w_i^d = tf_{id} * idf_i \tag{1}$$

$$tf_{id} = \frac{f_{id}}{\max_l(f_{ld})} \tag{2}$$

$$idf_i = \log \frac{N_d}{n_i} \tag{3}$$

where f_{id} represents raw frequency of term i in document d and $\max_l(f_{ld})$ is the frequency number of term l which has the maximum frequency in d. N_d is the number of documents in training set and n_i is the number of documents that have the same frequency number of term i as d.

For a interest subject s, the document set involved for the user u is represented as D_s^u. The weight of a specific subject s for user u can be calculated using the accumulated function:

$$w_s(u) = \sum_{d \in D_s^u} w_s^d * \eta(s, d) \tag{4}$$

where $\eta(s, d) = 1$ if $s \in d$; otherwise $\eta(s, d) = 0$.

Furthermore, content interest degree of subject s for user u is computed as below.

$$Cid_s(u) = \frac{w_s(u)}{\max_{s' \in S_u}\{w_{s'}(u)\}} \tag{5}$$

where S_u is the set of subjects the user u is interested in.

2) Semantic Interest Degree of Subject: To calculate semantic interest degree, we construct a target user's personal interest tree by the user's interest subjects. In addition, in order to better analyze semantic relations of subjects, we also consider hierarchical relationships, such as *is-a* and *part-of*. In an ontology tree, semantic specificity of subject was investigated based on the vertical hierarchical structure. The hierarchical affiliation of category "sport" in a domain ontology is illustrated in Figure 2.

The sematic specificity degree of target subject inherits from the parent node with different propagation strengths according to multiple levels and different number of siblings. However, semantic interest is reflected by semantic strength of subject relying on the user's behaviors. Semantic strength degree is the accumulation of sematic specificity degree. Semantic strength degree for the user u on subject s is calculated by:

$$Ssd_s(u) = \sum_{d \in D_s^u} Scd(s) * \gamma(s, d) \tag{6}$$

where $\gamma(s, d) = 1$ if $s \in d$; otherwise $\gamma(s, d) = 0$.

Taking into account all subjects the user is involved in, semantic interest degree of subject s for the user u can be computed as below.

$$Sid_s(u) = \frac{Ssd_s(u)}{\max_{s' \in S_u}\{Ssd_{s'}(u)\}} \tag{7}$$

where S_u is the set of subjects user u is interested in.

With the combination of two kinds of user interests, user model for a target user is represented in terms of content interest degree and semantic interest degree.

E. Satisfiability Checking of User Model

Given a $U_c = \{(U_c^1, Cid_1, Sid_1), (U_c^2, Cid_2, Sid_2), \cdots\}$ as the user model of a cloud service consumer extracted from domain ontology and real-world applications, and a composite service with a sequence of cloud services $\prod = (s_1, s_2, \cdots, s_m)$, where each cloud service $s = (I, O, Q, U)$. We use *Part-of* as the semantic expansion of each interest in a cloud service s_i, i.e., $s_i.U$. Therefore, semantic expansion interests $SemEx_i$ of the cloud service s_i can be accumulated by the children of each interest $U^j \in s_i.U \in \prod$.

$$SemEx_i = \bigcup_{U^j \in s_i.U \in \prod} \textit{Part-of}(po, U^j) \tag{8}$$

As a result, in conjunction with the interest features of each cloud service in a composition solution to a CSR problem, the combination of user model of a composite cloud service $\prod$ is computed by

$$U_{\prod} = \bigcup_{s_i \in \prod} (s_i.U \cup SemEx_i) \tag{9}$$

The satisfiability checking between U_c and $U_{\prod}$ is as follows. We first filter out those interest subjects from $U_c = \{(U_c^1, Cid_1, Sid_1), (U_c^2, Cid_2, Sid_2), \cdots\}$ where their corresponding content interest degree Cid and semantic interest degree Sid cannot satisfy a specified lower threshold. We denote the remaining interest subjects as $U_c^{'}$. Furthermore, relation of inclusion between each service interest $U^j \in U_{\prod}$ and each interest subject in $U_c^k \in U_c^{'}$ is iteratively checked for the user model matching. Finally, the composite cloud service could be recommended to the cloud service consumer, while the user model checking is satisfied between $U_{\prod}$ and $U_c^{'}$.

V. Empirical Experiment

To validate the feasibility of our proposed work for cloud service recommendation using service composition and user modeling, we have implemented two prototype systems in Java for dynamic composition of Web services and service platform based on user model, respectively. These two prototype systems can be plugged into the framework of cloud service recommendation as core components for personalized cloud service composition.

A. Service Composition Planning System

For the dynamic composition of Web services, we have implemented a prototype system based on a highly efficient automated planner Metric-FF and a our developed CSTE planner. The service composition system is outlined in Fig. 3. It has three main modules: service translation from WSDL to planning domain in PDDL, dynamically searching for a composite service considering QoS optimization using Metric-FF or CSTE planner.

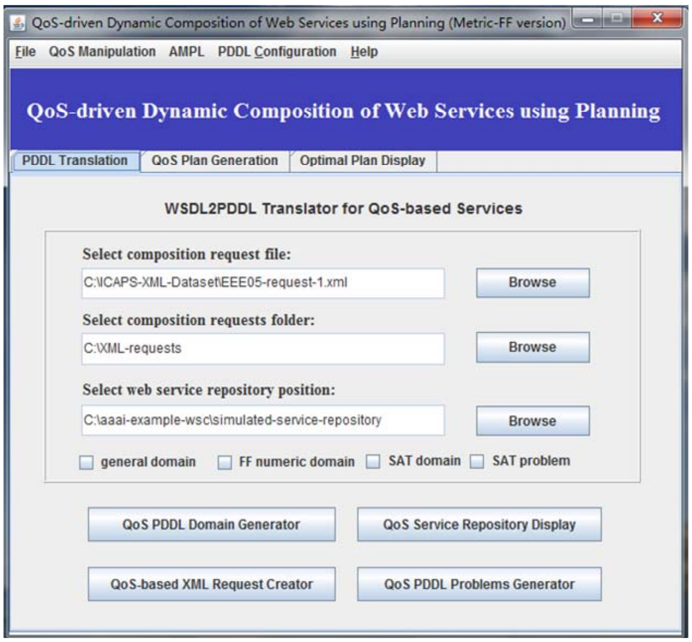

Figure 3. QoS-aware dynamic composition of Web services using planning.

During the process of dynamic composition of Web services, we apply different automated planners in terms of the requirements of service requesters. More specifically, when a service requester submits a composition request with all five cloud QoS constraints in Section III, we translate the service composition problem to a cost sensitive temporally expressive (CSTE) planning problem [6], [1], which is solved by our developed CSTE planner that not only takes logical reasoning and temporal planning into account, but also optimizes overall QoS of a found composite service. For a restricted class of QoS-aware service composition problems where there are no cloud QoS constraints on temporal restriction and average-based constraint, we can solve the Web service composition problem using an efficient metric-based planner (Metric-FF [5]).

B. Service Platform based on User Model

For the application of cloud user model, the fundamental experiments of user model construction have been conducted in our previous work [4]. A service platform based on user model was developed and empirical experiment was tested in this platform. In our experiment, the creation and update of user model were implemented at set intervals. For example, a user on the Internet likes sports, and one would keep special concerns on sports for long time, and also pay attention to the abundant news of sports from the websites. Our service platform can monitor user's behaviors and record relevant information, and then create corresponding user model for

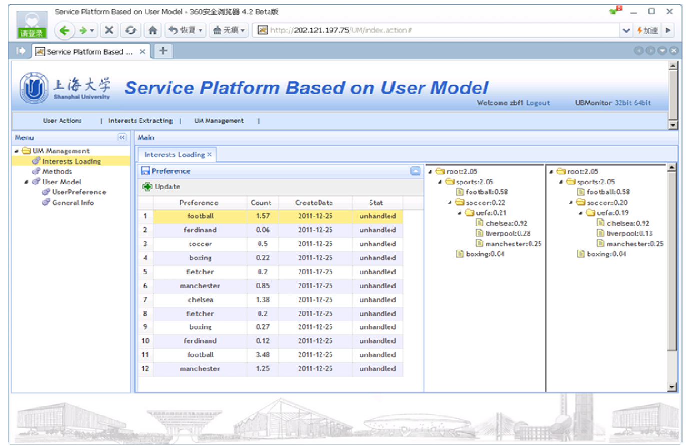

Figure 4. Service platform based on user model.

the target user according to historical behaviors. Figure 4 illustrates part of the user model, which extracts from a variety of user operations, including click action, scroll action, close action, and so on.

For a target user, as user's interests often drift, so the corresponding user model grows up with his browsing action being recorded automatically in the platform. Some preferences in one's user model increases while some will decrease contrarily, and some of them will disappear. The user model is visualized as tree structure, which contains previous interests and updated interests.

VI. Conclusions and Future Work

This paper proposes a novel framework for cloud service recommendation using the techniques of planning-based Web service composition and user modeling. The method first translates cloud services to a cloud service planning domain. Then, we integrate it together with a service request to model a cloud service planning problem. Finally, we invoke self-developed or highly efficient existing planners to generate a composite cloud service for the cloud service request. Our proposed framework could be implemented by the integration of Web service composition and user modeling prototype systems.

Our future work focuses on the improvement of user modeling checking, the integration of the service composition and user modeling systems, and the validation on personalized service recommendation in a real-world cloud platform.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61303096, 61300100), Shanghai Natural Science Foundation (13ZR1454600,13ZR1451000), an Innovation Program of Shanghai Municipal Education Commission (14YZ017), and a Specialized Research Fund for the Doctoral Program of Higher Education (20133108120029).

References

[1] G. Zou, Q. Lu, Y. Chen, R. Huang, Y. Xu, and Y. Xiang, "QoS-aware dynamic composition of Web services using numerical temporal planning," *IEEE Transactions on Services Computing (TSC)*, vol. 7, no. 1, pp. 18–31, 2014.

[2] M. Armbrust, A. Fox, R. Griffith *et al.*, "Above the clouds: A berkeley view of cloud computing," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*, 2009.

[3] K. Chen and W. Zheng, "Cloud Computing: System Instances and Current Research," *Journal of Software*, vol. 20, no. 5, pp. 1337–1348, 2009.

[4] J. Zheng, B. Zhang, and G. Zou, "Multi-granularity recommendation based on ontology user model," in *Proceedings of the IEEE International Conference on Cyber, Physical and Social Computing*, 2013.

[5] J. Hoffmann *et al.*, "The Metric-FF planning system: Translating ignoring delete lists to numeric state variables," *Journal of Artificial Intelligence Research (JAIR)*, vol. 20, no. 1, pp. 291–341, 2003.

[6] Q. Lu, R. Huang, Y. Chen *et al.*, "A SAT-based approach to cost-sensitive temporally expressive planning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 1, p. 18, 2013.

[7] J. H. Rao and X. M. Su, "A survey of automated Web service composition methods," in *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*, 2005.

[8] S. C. Oh, D. Lee, and S. R. T. Kumara, "Effective Web service composition in diverse and large-scale service networks," *IEEE Transactions on Services Computing (TSC)*, vol. 1, no. 1, pp. 15–32, 2008.

[9] X. R. Zheng and Y. H. Yan, "An efficient syntactic Web service composition algorithm based on the planning graph model," in *Proceedings of the IEEE International Conference on Web Services (ICWS)*, 2008.

[10] B. Yang and Z. Qin, "Composing semantic Web services with PDDL," *Information Technology Journal*, vol. 9, no. 1, pp. 48–54, 2010.

[11] E. Sirin, B. Parsia, D. Wu *et al.*, "HTN planning for Web service composition using SHOP2," *Journal of Web Semantics (JWS)*, vol. 1, no. 4, pp. 377–396, 2004.

[12] J. Hoffmann, P. Bertoli, M. Helmert *et al.*, "Message-based Web service composition, integrity constraints, and planning under uncertainty: A new connection," *Journal of Artificial Intelligence Research (JAIR)*, vol. 35, no. 1, pp. 49–117, 2009.

[13] N. Agarwal, M. Galan, H. Liu, and S. Subramanya, "Wiscoll: Collective wisdom based blog clustering," *Information Sciences*, vol. 180, no. 1, pp. 39–61, 2010.

[14] Q. Mei, C. Liu, H. Su, and C. Zhai, "A probabilistic approach to spatiotemporal theme pattern mining on weblogs," in *Proceedings of the International World Wide Web Conference (WWW)*, 2006.

[15] C.-H. Lee, "Mining spatio-temporal information on microblogging streams using a density-based online clustering method," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9623–9641, 2012.

[16] S. Garcia, P. O'Mahony, and B. Smyth, "Mining the real-time web: a novel approach to product recommendation," *Knowledge-Based Systems*, vol. 29, pp. 3–11, 2012.

[17] Y. Li, D. McLean, Z. A. Bandar *et al.*, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 18, no. 8, pp. 1138–1150, 2006.

[18] S. K. Ray and S. Singh, "Blog content based recommendation framework using wordnet and multiple ontologies," in *Proceedings of the International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 2010.

[19] S. E. Middleton, D. De Roure, and N. R. Shadbolt, "Ontology-based recommender systems," in *Handbook on ontologies*. Springer, 2004, pp. 477–498.

[20] L. Zeng, B. Benatallah, A. Ngu *et al.*, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering (TSE)*, vol. 30, no. 5, pp. 311–327, 2004.