PASS: An Approach to Personalized Automated Service Composition

Yang Li, JinPeng Huai, Hailong Sun, Ting Deng, Huipeng Guo
State Key Lab. of Software Development Environment
School of Computer Science and Engineering, Beihang University, Beijing, China
liyang@act.buaa.edu.cn

Abstract

With the rapid development of SOC (Service oriented computing)*, the automated service composition has become an important research direction. Through automated service composition, business processes need not to be constructed in advance, which helps to improve the flexibility of service composition. The current research on automated service composition is mainly based on AI techniques, and a common domain-oriented knowledge base is usually required to perform the heuristic planning. In practice, it is impossible for the knowledge base to characterize the personalized requirements of different users, so the AI-based methods can not apply to the user-centric application scenarios. In this paper, we propose PASS, a novel approach to personalized automated service composition. With PASS, both the hard-constraints represented by user's initial state, and the soft-constraints represented by user preferences can be satisfied in the process of automated service composition. Furthermore, three algorithms are designed to implement preference-aware automated service composition. In these algorithms, the Pareto dominance principle and relaxation degree are used to select the most satisfied composite service for users. Finally, comprehensive simulations are conducted to evaluate the performance and effectiveness of the proposed algorithms.*

1. Introduction

Along with the wide acceptance of SOC (Service-Oriented Computing), the amount of web services over the Internet has been growing continuously. As a result, service composition becomes a cost-effective method to create new value-added services[1]. Especially, automated service composition has been a promising approach to meet users' flexible requirements since it can avoid the limitations of static business processes.

The recent research on automated service composition is mainly based on the AI techniques [2-5]. In those efforts, a domain-specific knowledge base is first built for the reasoning process. Usually, such a knowledge base includes some heuristic rules serving as the common knowledge of a certain domain. With a knowledge base, the automated service composition is implemented through planning based on user's initial states and requirements.

In modern business environment, the ability to provide user with personalized services has become core competitive force for an enterprise. And recently user preferences also draw more and more attention in the service-oriented research field[6, 7]. However, the traditional automated service composition approaches can not meet of the personalized user requirements due to the following facts:

1) The common knowledge can not characterize the personalized requirements. For example, if a user wants to go to another city on vacation that is over 1000 miles away, then the best way for the user is to go there by airplane. And this is intuitively reasonable heuristic domain knowledge. However the user may be afraid of flying and can not accept the cost of flying as well, thus airplane is not the best way for him. As a matter of fact, for a specific user, his requirements are changeable and unpredictable in different contexts. Therefore it is impossible to describe all personalized requirements in a common knowledge base.

2) The user requirements not only include the hard-constraints, but also include the soft-constraints. In traditional approaches, only a user's initial state is considered as a constraint to implement planning, e.g., a user is VIP, and he has 500 dollars, and so on. But this kind of hard-constraint is not enough for describing user requirement in real business scenario. For example, a user wants to go to somewhere on vacation, and except his initial state, he has some preference, e.g., he prefers the fastest and the cheapest conveyance. Unfortunately, there may be no way to satisfy his preference at all. Thus, he has to make a compromise to select a feasible conveyance. This kind of soft-constraint is usually fuzzy and negotiable. As a matter of fact, user-centered service composition should pay much more attention to flexible user preferences for improving the user satisfaction. In addition, the soft-constraints will also improve the success rate of service composition thanks to the negotiable feature.

In this paper we propose an approach to implement personalized automated service composition. In this approach, we first introduce a user-preference-aware automated service composition model by extending

978-0-7695-3283-7/08 $25.00 © 2008 IEEE
DOI 10.1109/SCC.2008.24

HTN(Hierarchical Task Network) model[2, 3]. In this model, users' initial state and a set of preferences are considered as personalized user requirements. Then we design three algorithms to implement automated service composition to provide user with the most appropriate composite service based on Pareto dominance principle and relaxation degree. In the context of this work, we only consider the abstract composite service. In other words, we do not care how to select services from the candidate service instances. Thus, by default, we call a composite service as a plan, i.e., the automated service composition process is considered as a planning process. Finally, we design a set of experiments to evaluate the efficiency and effectiveness of these algorithms.

The rest of the paper is organized as follows: Section 2 provides detailed description of personalized service composition problem; in Section 3 we propose three algorithms to implement preference-aware automated service composition; Section 4 presents comprehensive experimental results through simulations; and in Section 5 we present the state of the art of related research; finally section 6 concludes this work.

2. Motivation example

2.1 Preliminary

HTN (Hierarchical Task Network) is an important technology used in automated service composition. A HTN is composed of a task network, primitive tasks, compound tasks and methods. In a HTN, the task planning is implemented through decomposing tasks. A task network is a collection of partially ordered tasks with constraints and data structure; a primitive task is an executable task; a compound task is composed of a group of primitive tasks; a method is a task network which describes how to obtain the primitive tasks.

SHOP2[2, 3] is a domain-independent HTN planning system. In the rest of the paper, we will describe our methods on the basis of SHOP2. For using SHOP2, a domain-dependent knowledge base is required to conduct service composition. The knowledge base is composed of operators and methods. An operator describes an action which can complete a primitive task, and usually represents a web service; a method describes how to decompose a compound task into a partial set of subtasks.

Definition 1 Operator. An operator can be represented as ($h(\vec{v})$*, Pre, Del, Add*):

- $h(\vec{v})$ is a primitive task with input parameters $\vec{v}$;
- *Pre* is the operator's precondition;
- *Del, Add* represent the execution consequences of the operator;

Definition 2 Method. A method can be described as ($h(\vec{v})$, Pre_1, T_1, Pre_2, T_2…):

- $h(\vec{v})$ is a compound task with input parameters $\vec{v}$
- T_i is a partially ordered set of subtasks
- Pre_i is T_i's precondition

Based on the above definitions, we call the set of T_i and their partially ordered relationship as a task decomposition tree in one task decomposition process.

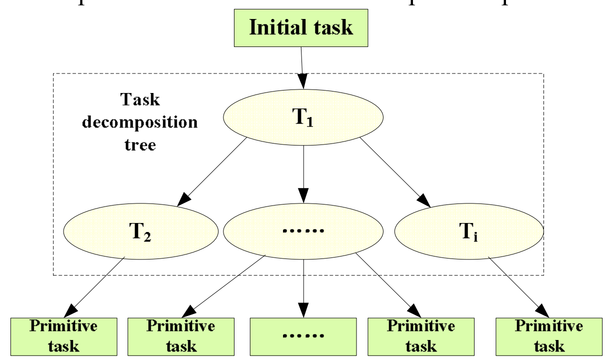

Figure 1. Task decompose tree

2.2 User preference

User preferences play an important role in business activities, and get more and more attention in web service research field recently, especially in personalized e-services[6][8]. The psychology research [9]shows that users expect to obtain the similar results in the interaction with service compared to the interaction with the human being. And no users expect that a composite service provider can predict their personalized requirements, thus users would rather provide their specific preferences.

In business environment, the amount of web services has been growing continuously, and many similar composite services can be constructed based on the existing services. But there still exists differences between these composite services, and these differences can be categorized into the following categories and be described by preference.

Functional differences. Some similar composite services have different function because they contain different component services. For example, two travel services, (*airplane, hotel, rent car*) and (*airplane, hotel*).

Behavioral differences. Different composite services may refer to different behavioral actions. For example, one composite service's precondition is that user must commit to pay before delivery; while another composite service requires a user to pay after delivery. Obviously, these two services need different preconditions, thus they can be applied to different users. When users want to purchase expensive goods, they usually prefer to use the latter service.

QoS differences. It is quite obvious that different composite services may deliver different QoS features. For example, a composite service using the DES algorithm is more efficient than a service using the RSA algorithm, whereas the former has less privacy than the latter.

Data differences. A composite service's preconditions may also include some data constraints. For example, a specific travel service can only apply to those users with certain destinations and transport options.

2.3 Motivation example

In this section, we consider a scenario in which a user in Beijing decides to take a vacation to Shanghai in five days, and he submits his requirement to travel agency so as to obtain an appropriate travel plan. This user's hard-constraints are his initial states, i.e., travel agency's ordinary membership and current date. In this example, the heuristic rules about how to find appropriate plans according to user's initial state have been deleted in order to avoid the mismatching domain description and personalized user requirement. In domain description, only the necessary rules are preserved.

The hard-constraints can be represented by SQL-like language:

SELECT Travel plan
FROM Domain description *D*
WHERE departure date='after 5 days'
AND Person＝'ordinary member'

According to user's initial state, we can conduct automated service composition based on *D*. As shown in Figure 2, because this user is not the VIP of the travel agency, and has enough time before departure, only the fixed routine method can be adopted. The fixed routine method consists of a few subtasks including transport and booking hotels etc. In these subtasks, transport and booking hotel subtasks are compound task, which need to be further decomposed. The transport subtask can be implemented by choosing airplane or train, and the hotel subtask can be implemented by choosing renewable-booking and *unrenewable-booking* hotels.

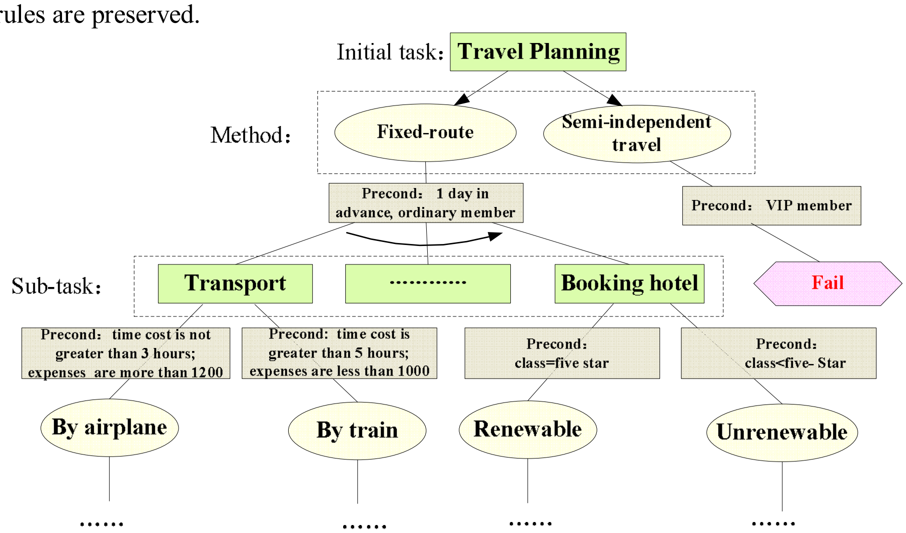

Figure 2. Travel agency example

Now, there still exist several different plans that satisfy user's hard-constraints. In this example, both booking hotel and transport subtasks can be implemented by two different methods. Because users' initial state do not refer to the specific requirements about time cost, expenses and hotel class, the preconditions of those methods are ignored automatically. Thereafter, in traditional approaches, a non-deterministic choice will be done despite of user's satisfaction. As a matter of fact, users would usually rather provide their specific preferences. Nevertheless, if a user want to provide his preferences exactly as (*time cost is no greater than 3 hours, expenses are less than 1000, class=five star*), there will not exist the kind of plan, so as to cause unavoidable failure of service composition.

To address the above issues, we notice that users usually do not expect to get a plan which satisfies all of his requirements. Therefore, we allow users to submit some soft-constraints besides hard-constraints, i.e. preferences, and select the most appropriate composite service for users. The extended SQL-like language is as follows:

SELECT Travel plan
FROM Domain description *D*
WHERE departure date= 'after 5 days'
AND Person＝'ordinary member'
Preference Lowest (timecost)
AND lowest (expenses)
AND Highest class (Hotel)

According to the soft-constraints, we know the user prefers to choose the plan that satisfies the precondition (*time cost is no greater than 3 hours , expenses are less than 1000 and class=five star*), However such a plan does not exist, therefore relaxation of the soft-constraints is required to find the most appropriate plan for the user. In this example, the user can choose the plan-*airplane+renewable booking* with compromise.

Thus, user-centered services composition should involve a high degree of respect for user preferences. Note that user preferences can not only be submitted by users, but also can be automatically extracted from user profiles or history records.

3. Preference-aware automated service composition

In this section, we present three automated service composition algorithms to provide user the best composite service based on preference, including P-global, R-global, P-local algorithm. Our algorithms are designed on the basis of SHOP2.

Because user requirements not only include hard-constraints, but also soft-constraints, so we first need to extend the model of the planning problem. All of algorithms proposed in this work are based on this model.

Definition 3 (preference). A preference p on variable x is a totally ordered set in domain D of variable x.

Definition 4 (planning problem). A planning problem is a quaternion (S, T, D, SR), where S is the initial state, T is the task list, D is a domain-specific description and SR is a preference set corresponding to variable set X. A plan is a sequence of operations (op_1, $op_2 ... op_n$) which can realize the task list T from the initial state S.

User requirements often involve a few atomic preferences, so we need to select the best plan for all of these preferences. In most of the time, it is difficult to find a plan which can satisfy the requirements of all the atomic preferences. In some research work[6, 7], a global utility function is implicitly or explicitly assumed to do trade-off between different atomic preferences. Nevertheless, the utility-based approaches are not convenient for users. For example, it can be difficult for a user to determine the weight of the cost and performance. Hence we incorporate the theory of Pareto dominance to select the best plans in the case that the atomic preferences are incomparable.

Definition 5 (Pareto dominance). Let A and B be the two candidate plans of planning problem (S, T, D, SR), where the preference set SR is composed of n preferences including $p_1 p_2,\ldots,p_n$, and p_i is the total order $\prec_{r_i}$ on the domain of variable x_i (i=1,2,…,n). Let the values of variables x_i (i=1,2,…,n) in plan A and B are ($v_{A1},\ldots,v_{An}$) and ($v_{B1},\ldots,v_{Bn}$), respectively, the plan A is said to Pareto-dominate plan B with respect to preference set SR, represented as $A \triangleright_{SR} B$, iff there exists j $(1 \le j \le n)$ such that $v_{Bj} \prec_{r_i} v_{Aj}$ and there does not exist i $(1 \le i \le n)$ such that $v_{Ai} \prec_{r_i} v_{Bi}$.

Definition 6 Pareto-dominant plan. Let A be a plan of planning problem (S, T, D, SR). if there is no plan B $(A \ne B)$ for the planning problem (S, T, D, SR) such that $B \triangleright_{SR} A$, we say A is a Pareto-dominant plan of planning problem (S, T, D, SR).

According to above definition, we can intuitively propose a primitive algorithm ***P-global***. This algorithm uses the SHOP2 algorithm to obtain all the plans satisfying (S, T, D), then randomly select one of the plans that are pareto-dominant.

Algorithm: *P-global*

Input: *s, T, D, SR,*
Output: *Plan*

```
Plan=Shop2(s,T,D)//search all the possible plans
for each p∈ Plan {
  for each p'∈ Plan {
    if (p ▷SR p' ) then Plan =Plan\ { p'}
    else {
      Plan =Plan\ {p}
       break
    }
}
    Plan=Non-deterministic choose p in Plan
    return p;
```

It is easy to know that the size of the Pareto-dominant plan set is exponentially increasing with the size of the preference vectors[10], which cause the complexity of the P-global algorithm to be high. In addition, the Pareto dominance principle can not completely reflect user satisfaction degree. As shown by the example in figure 2, *airplane+renewable* plan with (time cost is not greater than 3 hours , expenses are more than 1200, class is five star) and *train+unrenewable* (*time cost is greater than 5 hours , expenses are less than 800, class<five star*) are both pareto-dominant plan, but they have different user satisfaction degree. The former plan only can satisfy one preference, while the latter plan satisfies two preferences. Thus, we further propose the relaxation degree concept to describe the user satisfaction degree of a plan.

Definition 7 relaxation degree. Let preference p is a totally ordered set in domain D of variable x. For u, $v \in D$, if $u \prec_r v$, we say v dominates u with respect to p. For each sequence $v_{n-1} \prec_r v_{n-2} \prec_r \cdots \prec_r v_1 \prec_r v$ and there is no u in D such that $v \prec_r u$, we define the relaxation degree $rd(v_i)$ of v_i as i+1 and the relaxation degree of v is 1.

Definition 8 relaxation degree of a partially ordered set of subtasks. Let *Pre T* be a partially ordered set of subtasks, and p be a preference on the variable x_i, where $P(x_i{=}v_i)$ is a predicate in precondition *Pre*. The relaxation degree $rd(Pre\ T, x_i)$ of *Pre T* on variable x_i is defined as the relaxation degree of v_j. If *Pre T* has relaxation degrees $rd_1,\ldots,rd_n$ on variables $x_1,\ldots,x_n$, respectively, we say that the relaxation degree $rd(Pre\ T)$ of $Pre\ T$ is the sum of $rd_1,\ldots,rd_n$, i.e., $rd(Pre\ T)=\sum_{i=1}^{n} rd_i$.

Definition 9 relaxation degree of a plan. Let the task decomposition tree of plan A be composed of n partially ordered set of tasks $Pre_i\ T_i$ ($1 \le i \le n$), then the relaxation degree $rd(A)$ of A is the sum of the relaxation degrees of all the $Pre_i\ T_i$ ($1 \le i \le n$),i.e., $rd(A)=\sum_{i=1}^{n} rd\ (Pre_i\ T_i)$.

Theorem 1 The plan with the minimal relaxation degree is a Pareto-dominant plan.

Proof: Let A is a plan of planning problem (S, T, D, SR) with minimal relaxation degree. Assume that A is not a Pareto-dominant plan, and there is a plan B for the planning problem (S, T, D, SR) such that $B \rhd_{SR} A$. Let the variable set of preference set SR be $\{x_1,\ldots,x_n\}$, and the values of variables $x_1,\ldots,x_n$ in plan A and plan B be $(v_{A1},\ldots,v_{An})$ and $(v_{B1},\ldots,v_{Bn})$ respectively. Then there is no index i ($1 \le i \le n$) such that $v_{Bi} \prec_{r_i} v_{Ai}$, and there exists an index j ($1 \le j \le n$) such that $v_{Aj} \prec_{r_i} v_{Bj}$. Then we have $\sum_{i=1}^{n} rd(v_{Ai}) > \sum_{i=1}^{n} rd(v_{Bi})$. From Definition 7 and Definition 8, the relaxation degree of plan A and B are $\sum_{i=1}^{n} rd(v_{Ai})$ and $\sum_{i=1}^{n} rd(v_{Bi})$ respectively, thus it can be concluded that the relaxation degree of A is greater than that of B which is contradictory with our assumption. Therefore the theorem is proved.

From Theorem 1, we can see that the relaxation degree does not conflict with the Pareto dominance principle. Therefore we propose ***R-global*** algorithm in which we select the plan with the minimal relaxation degree instead of using Pareto dominance principle. The complexity of Pareto dominance principle is O (n^2) while the complexity of the relaxation degree is O (n), where n is the number of user preference vectors. As a result, the ***R-global*** algorithm greatly improves the performance compared with the ***P-global*** algorithm.

Note that if we can avoid finding all of the candidate plans satisfying hard-constraints, then the efficiency of the algorithm can be further improved. In the following we present an algorithm called ***R-local***. In this algorithm, the pruning technology is used in planning so as to only consider those plans that have the minimal relaxation degree. Then the amount of candidate plans decreases, and the complexity of R-local is greatly reduced.

Algorithm: *R-Local*

```
Input: S, T, D, SR,
Output: Plan
Plan=null
while (T ≠ null)
{
Get t∈ T(t is the first task in T)
if t is a primitive task then
  find the operator op of t
  add the op to the Plan
else
{
   get the partial sets PT of task T whose preconditions are satisfied by s
   for each t'∈ PT {
      compute the relaxation degree of t'
      }
   rd0=minimum of the relaxation degree in PT
   delete every partial set of task T which rd>rd0
   add the PT to the T
   R-Local(S, T, D, SR)
  }
}
get the plan p which has the minimum relaxation degree
Plan=p
return Plan;
```

Theorem 2 The plan obtained from the R-Local algorithm is a Pareto-dominant plan.

Proof: Let plan A is a plan for the planning problem (S, T, D, SR) obtained from *R-Local* algorithm. Assume that A is not a Pareto-dominant plan, then there is a plan B such that $B \rhd_{SR} A$. Let the variable set of the preference set SR be $\{x_1,\ldots,x_n\}$, and the values of variables $x_1,\ldots,x_n$ in plan A and plan B be $(v_{A1},\ldots,v_{An})$ and $(v_{B1},\ldots,v_{Bn})$ respectively. Then there is no index i ($1 \le i \le n$) such that $v_{Bi} \prec_{r_i} v_{Ai}$, and there is an index j($1 \le j \le n$) such that $v_{Aj} \prec_{r_i} v_{Bj}$. Therefore, for each i ($1 \le i \le n$), we have $rd(v_{Ai}) \ge rd(v_{Bi})$. For each subset $\{x_{i1},\ldots,x_{ik}\} \subseteq \{x_1,\ldots,x_n\}$, the sum of relaxation degrees of A on variable $x_{i1},\ldots,x_{ik}$ is not less than the sum of relaxation degrees of B on variable $x_{i1},\ldots,x_{ik}$, i.e., $rd(v_{Ai1})+\ldots+rd(v_{Aik}) \ge rd(v_{Bi1})+\ldots+rd(v_{Bik})$. According to the *R-Local* algorithm, plan A will be removed, which is contradictory with our assumption. Therefore the theorem is proved.

4. Experiment and evaluation

In this section, we present our experimental results of the proposed service selection algorithms. Because we have analyzed the complexity of ***P-global*** and ***R-global*** algorithm, so we don't conduct further experiments of ***P-global*** algorithm, and only ***R-global*** are considered to compare with other algorithms. In addition, we propose ***P-local*** algorithm to compare. The difference between ***P-local*** and ***R-local*** algorithm is that the former uses Pareto dominance principle, while the latter uses relaxation degree.

The experiments are conducted on a computer with a Pentium IV 1.61GHz CPU and 1GB memory. In addition, we implement all these algorithms based on the JSHOP2[11], and use two examples of JSHOP2, Logistics and FreeCell as the benchmark. We randomly change the values of the preference attributes between [1, 10] for all the experiments. The SHOP2 algorithm's function is only used to select one plan instead of finding all plans satisfying the user's initial state. Finally, because of the high complexity of searching all candidate plans with JSHOP2, the number of plans is set to be 30. For each experiment, we repeat 100 times and report the average.

4.1 Performance metrics

In the following, we define some performance parameters so as to evaluate the performance of the proposed algorithms.

1) Density of preference (DP). $DP=\frac{N_v}{N_t}$, N_v represents the total number of user preference vectors, and N_t represents the number of T_i in a task decomposition tree.

2) Satisfaction degree (SD).

$$SD= 1-\frac{\sum_{i=1}^{n} rd(v_{Ai})-\sum_{i=1}^{n} rd_{\min}(v_{Ai})}{\sum_{i=1}^{n} rd_{\max}(v_{Ai})}$$

, $\sum_{i=1}^{n} rd_{\max}(v_{Ai})$ and $\sum_{i=1}^{n} rd_{\min}(v_{Ai})$ respectively represent the Max and Min of relaxation degree of the candidate plans, and $\sum_{i=1}^{n} rd\ (v_{Ai})$ represents the relaxation degree of the selected plan.

3) Departure degree (DD)

$$DD=\frac{\sum_{i=1}^{n} rd(v_{Ai})-\sum_{i=1}^{n} rd_{r-global}(v_{Ai})}{\sum_{i=1}^{n} rd\ (v_{Ai})}$$

, where $\sum_{i=1}^{n} rd_{r-global}(v_{Ai})$ represents the relaxation degree of the plan obtained by the R-global algorithm, and $\sum_{i=1}^{n} rd\ (v_{Ai})$ represents the relaxation degree of the selected plan.

4.2 Experimental results

1) Execution time

Figure 3 and Figure 4 shows the execution time of different algorithms using the Logistics and FreeCell applications respectively.

As shown in Figure 3 and Figure 4, with the increasing of the density of preference, the execution time of all algorithms increases except the shop2 algorithm. For the algorithms proposed in this work, the execution time of R-local is the least while that of R-global is the most. In addition, it can be easily to observe that R-local algorithm' execution time is not so sensitive to the changing of the density of preference as P-local and R-global algorithm are.

2) Satisfaction degree

Figure 5 and Figure 6 plots the satisfaction degree obtained by using different algorithms.

In the process of service composition, it is possible that the plan with the minimum relaxation degree according to user's preference can not be found. Thus, the satisfaction degree shows the degree that an algorithm can meet the user expectation.

As shown in Figure 5 and Figure 6, with the increasing of the density of preference, the satisfaction degree of all algorithms keeps going down except the shop2 algorithm. In all algorithms, R-global always can get the best satisfaction and the R-local is worse than R-global, and the shop2 has the worst satisfaction degree. In both the Logistics and Freecell applications, the gap between R-global and R-local is always within 0.15.

3) Departure degree

The second experiment shows that R-global has the best satisfaction degree. In this experiment, we further evaluate the difference between these algorithms by observing the departure degree. Figure 7 and Figure 8 respectively shows the experimental results.

As shown in Figure 7 and Figure 8, with the increasing of the density of preference, the departure degree of all three algorithms keeps going down. In all algorithms, R-local can always get the minimal departure degree. Especially in Figure 8, when the density of preference is 8, the departure degree is almost equal to 0.2.

4.3 Analysis

According to the above experimental results, as the number of density preference increases, R-global steadily outperforms the other algorithms in satisfying user's requirement. However, the time cost of R-global is quite high. R-local has worse satisfaction degree than R-global, but the departure degree is very small as the density of preference increases. Especially, it has the least execution time. Thus, we can conclude that R-local algorithm should be the first choice in dynamic service composition, especially when the user preference is complicated.

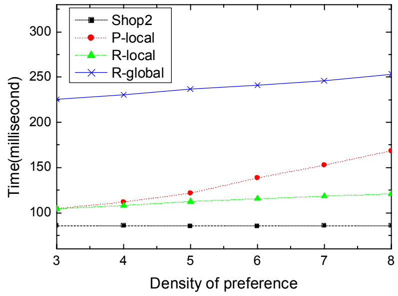

Figure 3 Execution time vs. DP

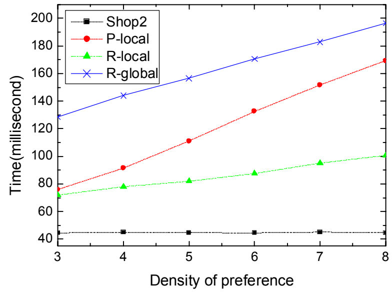

Figure 4 Execution time vs. DP

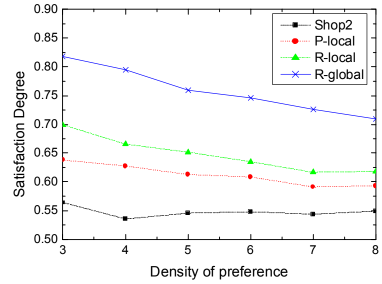

Figure 5 SD vs. DP

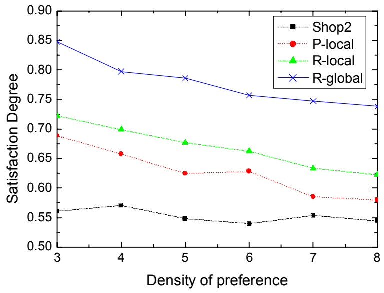

Figure 6 SD vs. DP

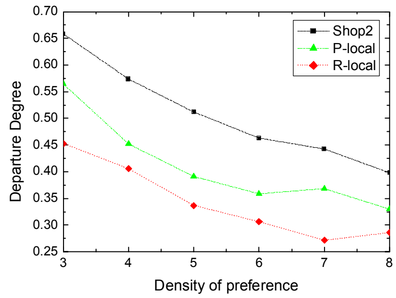

Figure 7 DD vs. DP

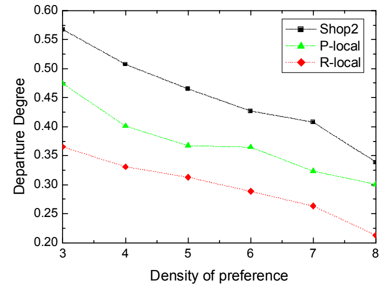

Figure 8 DD vs. DP

5. Related work

In recent years, automated service composition has become an active research field. Many approaches based on AI-planning techniques are proposed. An overview of these approaches can be found in [12, 13]. All of these approaches need a knowledge base to describe the common domain knowledge. Thus, it is difficult to deal with the personalized user requirements through using these approaches. In a word, these approaches focus on how to compose services automatically, but do not consider users' satisfaction with the generated composite services.

Actually, user preferences have been extensively studied in database [14, 15] and interactive decision [16, 17] research areas. With the development of SOA, user preferences also draw more and more attention in the field of service-oriented computing, but most of them mainly focus on the service selection instead of service composition.

In[18], a multi-attribute utility technique is adopted to model the user preferences. In [6], the authors address the selection of highly configurable web services. They also use a utility function to describe service configurations and relevant user preferences so as to attach price information to attribute values in a declarative manner. Then a method based on multi-attribute decision theory is proposed to select the optimal service.

These approaches have a common problem that users are required to provide the specific ranking information for aggregated functions of non-functional attributes, but it is difficult for user to provide that information in most cases.

In [8], the authors present a interactive personalized method to select services. In this approach, a user first provides hard-constraints declaratively, and gets the candidate service set that satisfy all the hard-constraints; then the user can extend his requirements according to the comprehensive input parameters of these candidate services; finally, the candidate services are sorted by the degree of satisfying the input parameters. In addition, this work mentions the selection of a plan using usage pattern, but no further explanations about how to do that are given. In [19], the authors also discuss the problem about how to find the most satisfying services for users.

The aforementioned research work mainly focuses on the service selection. In[20, 21], the user preferences are modeled as fuzzy rules to describe the ranking of composite services based on aggregation information of component services' non-functional attributes. Then, the user can get a list of candidate composite services sorted by rank, and choose a list according to a threshold of the rank of composite services. The above approaches mainly focus on selecting appropriate candidate services to construct optimal composite service, while our work addresses the issue that how to construct a business process which can best satisfy user requirement.

6. Conclusion

In this paper, we propose *PASS, a* novel approach to address the issue of how to satisfy the personalized user requirements in automated service composition. The major contributions of this research include: 1) We extend the HTN model to consider both the hard-constraints and the soft-constraints in user requirements so as to describe the user personalized requirements 2) Basing on this model, we design three algorithms to implement preference-aware automated service composition to provide user with the most satisfied composite service 3) We conduct comprehensive simulations to evaluate the performance and utility of the proposed algorithms.

The future work includes performing further research on the user preferences, and proposing a language to represent the comprehensive user requirements; further improving the performance of the proposed algorithms.

Acknowledgements This work was partly supported by the China National Science Foundation (No.90412011), National High Technology Research and Development Program of China (863 program) (No.2006AA01Z19A, No.2007AA010301), China 973 Fundamental R&D Program (No.2005CB321803) and National Natural Science Funds for Distinguished Young Scholar (No. 60525209).

References

[1] al., R.R.e. *The SAHARA Model for Service Composition Across Multiple Providers. Proc. of International Conference on Pervasive Computing (Pervasive 2002).*

[2] Wu D , P.B., Sirin E , et al. *Automating DAML-S Web services composition using SHOP2. Proceedings of 2nd International Semantic Web Conference (ISWC2003).*

[3] Sirin, E., et al., *HTN Planning for Web Service Composition Using SHOP2.* Journal of Web Semantics. 2004 (1): p. 377-396.

[4] Mark Carman, L.S., Paolo Traverso. *Web Service Composition Using Markov Decision Processes. Proceedings of the WAIM 2005*. 2005.

[5] Marco Pistore, P.T., Piergiorgio Bertoli. *Automated Composition of Web Services by Planning in Asynchronous Domains. Proceedings of the 15th International Conference on Automated Planning and Scheduling*. 2005.

[6] Lamparter, S., A. Ankolekar, and S. Grimm. *Preferencebased Selection of Highly Configurable Web Services. WWW*. 2007.

[7] Küster, U., et al. *DIANE An Integrated Approach to Automated Service Discovery, Matchmaking and Composition. WWW*. 2007.

[8] W.-T. Balke, M.W. *Towards personalized selection of web services. WWW*. 2003.

[9] Balke, W.-T. and M. Wagner. *Through Different Eyes-Assessing Multiple Conceptual Views for Querying Web Services. WWW*. 2004.

[10] Godfrey, P. *Skyline Cardinality for Relational Processing. Int Symp. on Foundations of Information and Knowledge Systems (FoIKS)*. 2004. Wil-helminenburg Castle, Austria.

[11]Maryland university. *JSHOP2. https://sourceforge.net/projects/shop*. 2005.

[12] Chan, M., Bishop, J., & Baresi, L. *Survey and Comparison of Planning Techniques for Web Services Composition. University of Pretoria Pretoria, Technical Report*. 2007: Polelo Research Group Department of Computer Science. South Africa.

[13] Peer, J. *Web Service Composition as AI Planning - a Survey. University of St., Gallen, Switzerland*. 2005.

[14] Chomicki, J. *Querying with intrinsic preferences. Proc. of the Int. Conf. on Extending Database Tech-nology (EDBT)*. 2002.

[15] Kießling, W. and G. Köstler. *Foundations of Preferences in Database Systems. Proc. of the Int. Conf. on Very Large) Databases (VLDB)*. 2002.

[16] Pu, P. *User-Involved Preference Elicitation*. in *Eighteenth International Joint Conference on Artificial Intelligence,Workshop on Configuration*. 2003. Mexico.

[17] C. Gonzales, P.P. *Graphical Models for Utility Elicitation. DIMACS/LAMSADE Workshop on Computer Science and Decision Theory*. 2004. France.

[18] Qainhui Althea Liang, J.-Y.C., Steven Miller. *Towards Semantic Service Request of Web Service Composition. Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE)*. 2005.

[19] Mahmoud, E.A.-M.a.Q.H. *Discovering the Best Web Service. WWW*. 2006.

[20] Sudhir Agarwal, S.L. *User Preference Based Automated Selection of Web Service Compositions. ICSOC Workshop on Dynamic Web Processes*. 2005.

[21] Lin M S, X.J.S., Guo H Q. *Solving QoS-driven Web service dynamic composition as fuzzy constraint satisfaction. The IEEE International Conference on e-Technology , e-Commerce and e-Service*. 2005.