

# Preference-Based Web Service Composition: Case-Based Planning Approach

**Yamina Hachemi, Sidi Mohamed Benslimane**

EEDIS Laboratory, Djillali Liabes University of Sidi Bel Abbes, Sidi Bel Abbes, 22000, Algeria

Email: {hachemi.amina, benslimane}@univ-sba.dz

**Abstract**— Web service selection is an indispensable process for web service composition. However it became a difficult task as many web services are increased on the web and mostly they offer similar functionalities, which service will be the best. User preferences are the key to retain only the best services for the composition. In this paper, we have proposed a web service composition model based on user preferences. To improve the process of web service composition we propose a case-based planning approach with user preferences which uses successful experiences in past to solve similar problems. In this paper we integrate user preferences in the phase of selection, adaptation and planning. Our main contributions are a new method of case retrieval, an extended algorithm of adaptation and planning with user preferences. Results obtained offer more than a solution to the user and taking both functional and non-functional requirements.

**Index Terms**— Web Service Composition; Case Based Planning; User Preferences.

## I. INTRODUCTION

Service-oriented architectures, especially Web services, enable access, discovery and the use of any application on the web using open standards XML [1], SOAP [2], WSDL [3] and UDDI [4] and Internet transport protocols. If a single service cannot satisfy the functionality required by the user, it is necessary to compose existing services together in order to fulfill the request.

The composition object is to combine the functionality of several Web services within a business process in order to respond to complex applications that single service could not meet. However, the composition of services is far from being a trivial task.

Web service composition is a hard and complex task, and out of the human capability to deal with the whole process manually. One of the factors of the complexity is to find appropriate services to be composed. However web services composition is usually based on the functional properties only neglecting a very important point which is the user preferences.

As the number of web services increases significantly during the recent years and therefore a huge repository of Web services to be searched, user preferences are used to select appropriate services. For example planning a travel, there will be several compositions of web services that allow the user to travel from a source location to a destination but such travel planning have to take into account the user preferences such as the total cost of the

trip, preferences on particular transportation companies or hotels, and times/dates for the travel.

Based on the assumption that the world is regular, so similar problems have similar solutions; As a consequence, solutions for similar problems are a useful starting point for new problem-solving.

Case-based planning (CBP) is a problem-solving method that uses a library of cases, where a case associates a past problem and goal description with a plan that solves the problem by achieving the goal [5]. In similar situations, CBP can take advantage of previous planning experiences by reusing stored plans.

In this paper, we present a novel approach to compute the best service compositions based on user preferences and CBP. The proposed approach differs from the previous works in that user's preferences are taken into count as an additional input, so we model preferences and incorporate them into the user request. We use an algorithm to determine the relevant services that may be used to answer the composition request. We propose a new composition with user preferences if the request is not solved before and our system is learning from experiences.

Our approach integrates the user preferences in the process of Case-Based Planning, the multi plan adaptation and the planning. We generate the plan for the user's new request by finding a composition plan from the library of cases. The newly generated plan together with the new request can now be stored as a case in the case base for future reuse. Many CBP retrieve and adapt a single plan [6] [7]. We propose to retrieve multiple plans using an algorithm for retrieving based on two steps find and select according to the preferences. The sub-plans retrieved are merged and adapted to solve the problem.

This paper is organized as follows: in section 2, we present an overview of recent methods that uses case-based reasoning and planning in the Web service composition process, in section 3, we introduces the case based planning approach and define user's preferences. in section 4, we present our approach for the service composition based on CBP using the user's preferences. Section 5 introduces an experimental methodology of our approach and finally, we present conclusions and future works in section 6.

## II. RELATED WORK

Artificial Intelligence techniques can provide a solution to the problem of service composition. In particular, there have been several proposals using AI planning.

Case-based reasoning (CBR) [8] is a problem solving methodology based on reutilizing specific knowledge of previously experienced and concrete problem situations (cases). Case-based planning is the application of the CBR methodology to planning, and as such, it is planning as remembering [9].

Different approaches based on CBR and CBP have been proposed for the composition of Web services. In this section, we review selected works based on their relevance for our approach.

Limthanmaphon et al. [10] are the first who proposed a framework for WS composition using the CBR technique for the discovery process. However, his similarity assessment method is based on keyword matching, so the accuracy is very low. The algorithm for retrieving a case is only based on the service name and this approach does not use the semantic descriptions to make efficient reasoning. So the automation aspects for Web services search and composition is affected.

Lajmi et al. [11] presented an approach semi-automated Web Service composition based on CBR technique and the semantic description of Web Services called WeSCo\_CBR. But their approach can only return the plan of the most similar case and leave the plan adaptation task to the user, and the similarity computation algorithm only considers the Input/Output signatures of the activities without considering the preconditions and effects.

Web service execution experiences are captured as cases by Thakker et al. [12], they use these cases for finding a solution for new problems. Use of ontologies also makes the framework extensible and reusable, but their method merely deals with the problem of similarity assessment.

Cheng et al. [13] proposed an approach based on service case adaptation. However, they do not solve the problem of inefficient retrieval existing in the case base with large capacity.

Liu et al. [7] used an indexing technology to organize the case base and present a case adaptation strategy for web services. However, they cannot offer solutions in the absence of previous similar cases.

Sun et al. [14] proposed a case based web services reasoner (CWSR). It mainly consists of an interface agent, a global web service base (GWB) and a CBR inference engine (CBRIE).

Hu et al. [6] generated the plan for the user's new request by automatically adapting the existing plan for the most similar past request. Afterward they put forward this method of web service composition with CBR, but their approaches merely deal with the problem of a case that have no matching in the case base.

Lee et al. [15] construct a goal model and extract intent from service requests, focusing on intention satisfaction and merging internal and external services to

meet user needs. They only use the substitution in the phase of adaptation.

Most of the approaches cited above, address the case representation, the retrieving process and the adaptation phase. However, they merely deal with the user preferences in these phases. We present here after our approach for planning semantic web service composition. The proposed approach is based on planning with user preferences for semantic web services composition. It assumes that both functional properties and non-functional ones are used in the composition, which ensures the fulfillment of user's requirements.

## III. BACKGROUND

### A. Case Based Planning

A case-based planning system [16] solves planning problems by making use of stored plans that were used to solve analogous problems. CBP is a type of case-based reasoning, which involves the use of stored experiences (cases). When a CBP system solves a new planning problem, the new plan is added to its case base for potential reuse in the future. Thus, we can say that the system learns from experience. In general, the following steps are executed when a new planning problem must be solved by a CBP system:

- Plan Retrieval: to retrieve cases from memory that are analogous to the new problem.
- Plan Evaluation: to evaluate the new plans by execution simulated execution, or analysis and choose one of them.
- Plan Adaptation: to repair any faults found in the new plan.
- Plan Revision: to test the solution new plan for success and repair it if a failure occurs during execution.
- Plan Storage: to eventually store as a new case in the case base.

### B. User Preferences

In web service composition the aim is to generate a solution that not only achieves some absolute goals and any constraints associated with them, but also is desirable with respect to some user preferences [17].

We argue that user preferences are a critical and missing component of most existing approaches to Web service composition. User preferences are the key for many reasons [18]:

- The introduction of preferences in queries provides a basis for rank-ordering the retrieved items, which is especially valuable in case of large sets of items satisfying a query.
- User preferences enable user to specify properties of solutions that make them more or less desirable. The composition system can use these to generate preferred solutions.
- The reason why user preferences are critical to Web service composition is with respect to how the composition is performed. A key component of

Web service composition is the selection of specific services used to realize the composition.

Motivating Scenario: Travelling planner

In the travel scenario a user is looking for a service that can take a travel request as input and as result produce its corresponding Book Flight, Book Hotel, Rent Car and Conference Register. The user prefers travel

with star alliance airplane. Suppose that the system have solved similar past requests.

We will consider that the first request produce the same services as the third request but with preferences. In our case we will take the third one. The second request produce three services .The planner will determine the execution order of these tasks. The table 1 shows the new problem with the retrieved cases.

Table 1. The new request and the two past requests

Request	Input	Output	Goal	User preferences
User query	Travel request()	Travel planning for conference attendance	BookFlight BookHotel RentCar ConferenceRegister	Star alliance airplane
Request 1	Travel request()	Travel planning	BookFlight BookHotel RentCar	
Request 2	Travel request()	Conference attendance	BookFlight BookHotel ConferenceRegister	
Request 3	Travel request()	Travel planning	BookFlight BookHotel RentCar	Star alliance airplane

When computing the similarity between the new request (travel planning for conference attendance) and the cases in the case base, the system has not obtained a direct solution. However, by decomposing the goal it obtains two solutions. The first contain the travel planning and the second the conference attendance. The system merges the solutions of case 1 and case 3 to obtain a solution to the problem. Finally, the plan and the request formed a new case, the system update the case base.

IV. PROPOSED APPROACH

Our approach called Case-based planning with preferences for web services composition (CBP-P). It is based on the use of two AI techniques CBR and planning. CBP to minimize the search time if the solution exists and improve continually learning of new cases and Planning with preferences if the problem cannot be solved by CBP based on the case base; the planner takes over and finds a composite service for the user.

The proposed approach consists of the following components as shown in Fig. 1.

1. Service request: users that need web services send requests.
2. Translation: it is the process of translating the user request. The purpose of the translation is to translate the user requests into a suitable manner (problem) manageable by the system. New problems are obtained through the translation.
3. Retrieval: in this phase all the possible solutions of the new problem are retrieved from the case base using a similarity assessment method and user preferences.

4. Planning: it consists to compose a new web service using the new problem and preferences.
5. Adaptation: the solutions obtained are adapted to solve the new problem.
6. Execution: the solution obtained is executed.
7. Learning: a new case is composed of the new problem and the solution obtained, the system decides if the new case will be added to the case base or not.

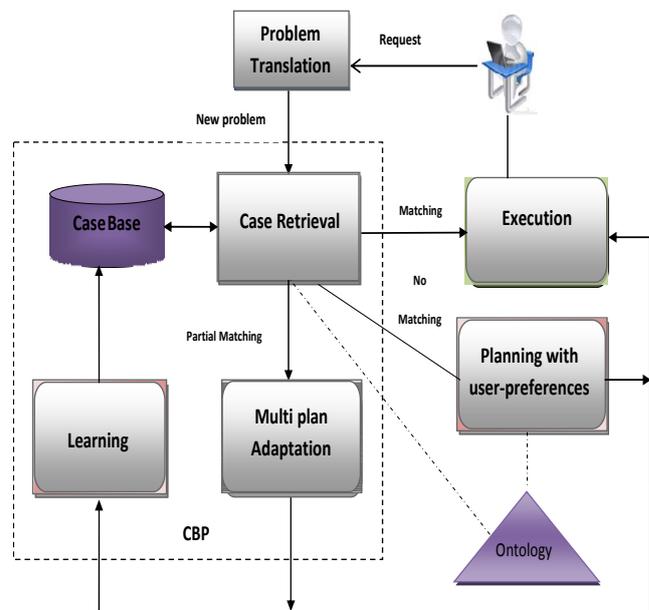


Fig. 1. CBP-P for WSC

A. Case Representation

The use of CBR requires the identification of a case, which needs to be represented by a model adapted to our problematic. This modeling allows us to describe each

component of a case. For the search of similar cases and the selection of the relevant cases, we need the case similarity computation and search procedures. A case is used to store the experience of solving a past problem; it always has the same components including a problem, a solution and possibly an evaluation.

In the context of service composition, the user's request can be seen as a problem description and the plan that fulfils the request is the solution to the problem.

#### Definition 1: Case

A case is a quintuple  $C = (PB, S, Q, N)$  where:

- $PB$  is the problem and represent the user's past request where:

$$PB = (I, O, T, SG, A)$$

- $I = \{i_1, i_2, \dots, i_n\}$  is a list of semantic data types stating what data are available at the beginning of the required service.
- $O = \{o_1, o_2, \dots, o_m\}$  are a list of semantic data types stating the required outputs after the execution of the required service
- $T$  is the Initial state
- $SG$  is a set of goals.
- $A$  represent the user preferences
- $S$  is the solution of the problem that represents a partial order plan.

$s = \{o_{p1}, o_{p2}, \dots, o_{pm}\}$ , that is, a sequence of instantiated operators that will achieve  $SG$  from  $S$ , with the best quality and respect the non-functional aspects.

- $Q$  isa set of attributes that represent the QoS (i.e., response time, cost, availability, reliability, etc.).
- $N$  count how many times the solution of the case has been successfully used to fit the new requests ( $NQ$ ).

Currently, Web services are mostly described by OWL-S since it supports effective automation of various Web services related activities including service discovery, composition, execution, and monitoring. An OWL-S description consists of three parts: service profile, service model and service grounding. We use the service profile to represent the inputs and outputs because it contains some description of the service.

#### Definition 2: Case base

A case base is a finite set of cases denoted by:

$$CB = \{C_1, C_2, \dots, C_n\}$$

Where the element  $C_i$  of  $CB$  is called Source case.

#### Definition 3: Plan

A. plan  $PL \{a_1, a_2, \dots, a_n\}$  is a set of sequences of actions, where  $a_i$  represent the action.  $PL$  is the solution to the problem  $NQ$  if all possible executions of the plan, starting from  $S$ , satisfy  $G$ .

#### B. Problem Generation

This step consists in the transformation of the introduced request into a planning problem, i.e. a suitable

manner that will be treated by the system. The user interface is used by a novice user who does not know Web service technologies. One of the most user-friendly and convenient interfaces to invoke services is a natural language interface. Several methods concerning natural language interfaces for Web services have been proposed [19, 20, and 21].

The following example illustrates a transformation of a user's natural language request into a planning problem. Notice that this mechanism is out of scope of this paper.

#### Example

Let the user's complicated request "For conference attendance, I want to book a Flight from Algiers to Paris then rent a car in Paris and reserve a hotel suite room from 23rd November to 25th November. I prefer to fly with a Star Alliance carrier".

This natural language request is transformed by the system into a planning problem  $PB = (I, O, S, SG, A)$ , where:

$$I = \{\text{Travel request}\}$$

$$O = \{\text{Travel planning for conference attendance}\}$$

$$S = \{\}$$

$$SG = \{\text{BookFlight, BookHotel, RentCar, ConferenceRegister}\}$$

$$A = \{\text{preference p1 (sometime (initiate (book-flight Star Alliance carrier)))}\}$$

#### C. Case Retrieval

To support large and growing case bases, a CBP system need scalable retrieval technique. Retrieval is the process of finding and getting an appropriate stored case, which is close to the requirements. The goal of retrieval is to find cases, which have the potential to be most useful. Case retrieval requires a combination of searching and matching. Similarity measurement is used as a tool to find the closest case which matches the query.

Whenever a user has a request, the service case base is searched to find the case which description part is similar to the user's request. The main module in this process is the similarity computation between each case and the new problem.

Given a new problem (request), two processes in the phase of retrieval are involved in our system. First, the so-called search process searches for cases that match the target problem. Then the so-called select process has in charge to choose the most appropriate source cases according to the user preferences in order to find an approximate matching. The source plans resulting from the retrieval process is adapted to build a solution to the target problem.

#### Plan search:

The problem generated from the past step is the input of the plan search algorithm; we have the sets of inputs (I), outputs (O), initial state (T), goals (SG) and user preferences (A). In this step we search a matching for the problem. If there is an exact matching, the system copies the solution for execution. Otherwise we have to search all possible solutions for a goal or a sub-goal in the case

base. If the set of candidate plans is empty we have to decompose the goal to sub-goals. When a goal is reached we store the goal and the solutions. We repeat this process until the set of goals (U) is empty and the algorithm 1 return a set of candidate plans to the next step. If the set of plans (SP) is empty, it means that no solution is found in the case base. The system has to compose a solution using a planner. In this step a planning with preferences is called to obtain a composition.

---

**Algorithm 1: plan search**

---

**Inputs:**  
 -  $PB = (J, O, T, SG, A)$  /\* New problem \*/  
**Outputs:**  
 -  $SP$  /\* set of plans \*/  
 $CP \leftarrow \{\}$  /\* set of candidate plans is empty \*/  
 1: Search (P) /\* find all the plans in the case base \*/  
 2: **If** ( $CP = \{\}$ ) **then**  
 3: Decompose (G) /\* goal decomposition \*/  
 4: **Else**  
 5: Select (P) /\* select plans \*/  
 6: Add (P,  $A_p$ ) to SP /\* add the select plan to the SP \*/  
 7:  $U = G - A_p$   
 8: **End if**  
 9: **Until** ( $U = \{\}$ )  
 10: **Return** (SP)  
 11: **If**  $SP = \{\}$  **then** /\* set of plans is empty \*/  
 12: Planning (NQ)  
 13: **Else**  
 14: Plan selection (SP)  
 15: **End if**

---

**Plan selection**

After the search step a selection is needed to obtain the best plans for the next step. Inputs for the plan selection algorithm 2 are set of goals (SG), set of potential plans (SP) and preferences (A). Solutions are selected according to user's preferences. The algorithm returns a set of preferred plans.

---

**Algorithm 2: plan selection**

---

**Input:**  
 $SG, SP, A$  /\* goals, set of potential plans, preferences \*/  
**Output:**  
 SPP  
 1: **For each** (G) from (SG)  
 2: Find plan (P) that match preferences (A)  
 3: **If found matched** (P) **then**  
 4:  $SP = SP - G$   
 5: **End if**  
 6: **End for**  
 7: **Return** (SPP)

---

Preferences represent user properties: When several services are available to perform the same activity, their properties such as total cost, choice of the airplane companies or hotels, dates and time become important in the selection process. In order to reason about preferences in web service, a model is needed to capture

the descriptions of these properties from a user perspective. Preferences are expressed using the language PDDL.

Once we know the user's preferences, we must relate them to preferences over plans. In particular, we must formally define when one plan is preferred to another.

We are looking for a most-preferred plan for a goal (sub goal). To define the notion of a most-preferred plan, we use an ordering relation that specifies when a plan is preferred to another.

The retrieval phase terminates with SP, the set of plans selected. If SP is empty, this means that no plan was found in the case base. In this case, we try to found a solution from scratch.

Many case based planning systems can only return a single old plan. This plan may achieve a subset of the goals and thus require extensive adaptation to achieve the remaining goals. The retrieval phase can return multiple plans when merged are the basis of the new plan.

**D. Planning with preferences**

In this section, we describe the web service composition using planning with preferences. If the result of the retrieval phase is not successful and the system cannot find a similar case in the case base, the system proceeds to a planning from scratch.

For specifying user preferences we use the preference language Domain Definition Language PDDL3 [22]. The syntax of the preference language extends PDDL2.2 by enabling the specification of preferences and hard constraints. It also provides a way of defining a metric function that defines the quality of a plan.

Modeling user preferences is largely based on the recent planning language PDDL3 [23] that allow incorporating user preferences in planning problems. In PDDL3, preferences are described as logical assertions over states and state trajectories by defining basic preferences and temporal preferences.

Syntax: (preference [name] <GD>)

Example:  $A = \{p1, p2, p3, \dots\}$

(and (preference p1 (always (clean truck1))))

(preference p2 (and (at end (at package2 paris))))

(sometime (clean track1))))

(preference p3 (...))

... )

There are several approaches for automatic planning web service composition with user preferences that can be found such as HPLAN-P [24], SGPLAN [25], SCUP [26] and HTNPlan-P [27]. For this step we use an existing tool which takes the user preferences in the composition of web services. The planner SGPlan is the winner of the 5th International Planning Competitions [28]. SGPlan partitions a planning problem into sub problems, each with a goal state, and finds a feasible plan for each. Its contribution focuses on optimizing goal preferences for satisficing (sub-optimal) planning subtrack.

SGPLAN5 is the new version of the SGPLAN planner that also participated in IPC4. Features in the new version include a new heuristic, similar to the causal

graph heuristic used by FAST DOWNWARD, for planning at the subgoal level, and extensions to handle all the new features of PDDL3.0. Like several other planners, for problems with preferences SGPLAN5 employs a strategy of iteratively searching for better plans after the first plan has been found. Unlike other competing planners, however, trajectory constraints are not compiled away but handled directly by the search. Fig. 2, shows a part of the plan description

```

; Time 0.02
; ParsingTime 0.00
; NrActions
; MakeSpan
; MetricValue 4.000
; PlanningTechnique Modified-FF(best-first search) as the subplanner

0.001: (DRIVE TRUCK1 L4 L3) [355.600]
355.604: (LOAD PACKAGE4 TRUCK1 A2 L3) [1.000]
356.607: (LOAD PACKAGE5 TRUCK1 A1 L3) [1.000]
357.610: (DRIVE TRUCK1 L3 L1) [554.300]
911.913: (UNLOAD PACKAGE5 TRUCK1 A1 L1) [1.000]
912.916: (DELIVER PACKAGE5 L1) [1.000]
912.916: (UNLOAD PACKAGE4 TRUCK1 A2 L1) [1.000]
913.919: (DELIVER PACKAGE4 L1) [1.000]
913.919: (DRIVE TRUCK1 L1 L4) [880.500]
1794.422: (LOAD PACKAGE7 TRUCK1 A2 L4) [1.000]
1795.425: (LOAD PACKAGE8 TRUCK1 A1 L4) [1.000]
1796.428: (DRIVE TRUCK1 L4 L2) [1034.100]
2830.531: (UNLOAD PACKAGE8 TRUCK1 A1 L2) [1.000]
2831.534: (DELIVER PACKAGE8 L2) [1.000]
2831.534: (UNLOAD PACKAGE7 TRUCK1 A2 L2) [1.000]
2832.537: (DELIVER PACKAGE7 L2) [1.000]

```

Fig. 2. A part of the plan description

### E. Multi Plan Adaptation

The use of multiple plans during the adaptation phase is one of the interesting aspects of the case-based planning paradigm. A single past case that is similar to the complete new situation may not be found in complex planning situations with multiple goals. However, Planning can then be viewed as the process of merging and adapting these multiple complementary planning cases if several planning cases maybe found that cover independent subparts of the new planning situation in a complementary way.

The most common adaptation approaches cited in the literature to reuse past experience are: substitutional adaptation, transformational adaptation, and generative adaptation [29].

In our work we will focus on transformational adaptation techniques, particularly in search-based approaches. These techniques are used in CBR systems applied to planning domains [30]. Search-based approaches to reuse are, often, limited to reusing a solution from a single case. Compositional adaptations are reuse techniques that find new solutions from multiple cases.

Although many plan adaptation methods are suitable for our task, we use the MPA algorithm because it is complete and systematic in compare with other adaptation algorithms in the literature.

MPA (Multi-Plan Adapter) [31] is an extension of SPA (Systematic Plan Adaptor) [32] which allows for

reusing multiple plans. MPA breaks the different plans into smaller pieces, which are then recombined together.

For the MPA algorithm, it takes as input a partial plan and the case library to search partial plans. Our extended algorithm E-MPA takes the selected plans obtained from the retrieval phase.

---

#### Algorithm3: Extended Multi Plan Adaptation

---

##### Inputs:

$-P, SPP$  *\*/plan, set of preferred plans \*/*

##### Output:

$-P'$  */\*preferred plan\*/*

1:  $P' \leftarrow \text{Copy-Plan}(P)$

2:  $igs \leftarrow \text{GetIntermediateGoalStatement}(P')$

3:  $plan \leftarrow \text{RetrieveBestPlan}(SPP, igs)$

4:  $\{clipping, mapping\} \leftarrow \text{FitPlan}(plan, igs)$

5: *for cgp in mapping do*

6: *if Producer-Exists(oc-gl-pair, P')*

7: *then Splice-Link(oc-gl-pair, P', clipping)*

8: *else Splice-Step(oc-gl-pair, P', clipping)*

9: *AddNewOpenCond-GoalPairs(mapping, P')*

10: *Endif*

11: *Endfor*

12: *return (P')*

---

After adaptation, planning or exact matching, the execution binds the composite service activities to concrete Web services and returns the resulting composite service to the user.

## V. IMPLEMENTATION

To implement the proposed technique we used the JAVA programming language. We have also used PDDL3.0 [23] for representing the user preferences. We use domain-independent planning tool with user preferences SGPlan [27] as our planner.

### A. Methodology

The methodology of proposed model is given as:

1. User send request for a web service
2. Translation converts the user request into a new problem.
3. The new problem is compared to the case base. The system search solutions to the problem using a retrieval mechanism composed of two phases search and select using the user preferences. There are three possibilities: exact, partial or no matching.
4. If the system finds the exact solution then it sends it to the execution.
5. If there is not a matching in the case base, the system will use the planner to compose a new web service for the user request according to his preferences.
6. If the system finds solutions that must be adapt to the user request, adaptation phase is needed.

### B. Domain

We use the Trucks Domain which is a logistics domain about moving packages between locations by trucks under certain constraints. The loading space of

each truck is organized by areas: a package can be (un) loaded onto an area of a truck only if the areas between the area under consideration and the truck door are free. Moreover, some packages must be delivered within some deadlines. In this domain, it is important to find good quality plans.

The **trucks** problems sets comprised 20 problems. The number of preferences in these problem sets ranged in size, with several having over 100 preferences per problem instance.

*Goals:*

(delivered package1 I1)

(delivered package2 I2)

(delivered package3 I2)

*Preferences :*

(preference p1B (sometime-before (delivered package2 I2) (delivered package1 I1)))

(preference p4A (within 919.7 (delivered package1 I1)))

(preference p4B (within 919.7 (delivered package2 I2)))

(preference p4C (within 1813.7 (delivered package3 I2)))

Fig. 3, shows the mean successful percentage of our method of retrieving cases by decomposing the main goal to sub goals. It can be shown that when the number of goal increases, the percentage of successful retrieval also increases.

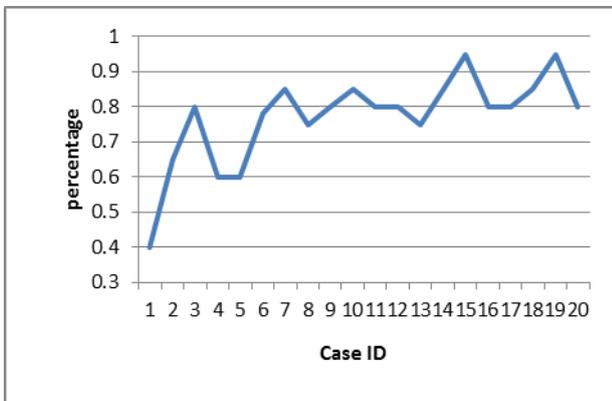


Fig. 3. Experimental results about successful service retrieval percentage

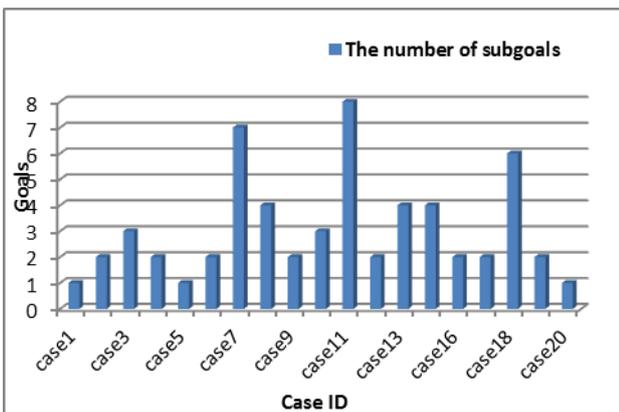


Fig. 4. Experimental results about found cases

C. Evaluation

Fig. 4, Shows the relationship between the number of cases found by decomposing the goal. In addition, it can be shown that the use of the retrieval method of searching cases can make the efficiency of web service composition higher than that of method based on searching the whole problem.

Fig. 5, Shows selected cases according to the user preferences. We can find several solutions that respond to the goal but not the user preferences. It can be shown that the selection method can give better solutions.

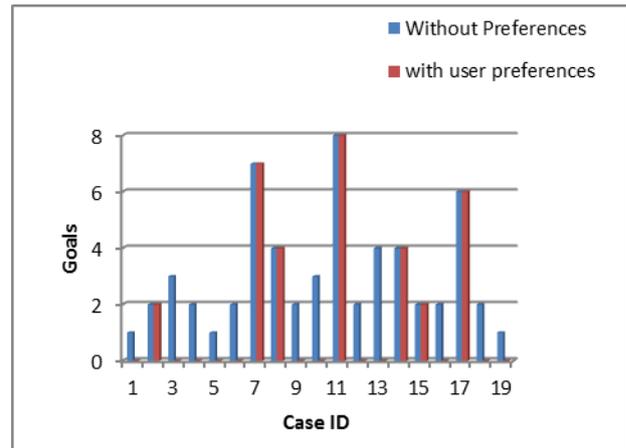


Fig. 5. Experimental results about selected cases

VI. CONCLUSION

We discussed in this paper the problem of web service composition. We first present outlined user preferences and case-based planning approach. We then described some existing work combining web service composition and case-based reasoning. We finally presented our proposal using a case-based planning process to support the composition of semantic web services according to the functional and non-functional properties (user preferences). It includes case representation, retrieval algorithm, case adaptation and planning.

We add the user preferences at the user request in the representation of the case. As to case retrieval, we used the retrieval algorithm with two main steps search and select to obtain best solutions from the case base. At the first time, we search cases closer to the problem than we select solutions according to the user preferences. In the procedure of service case adaptation, we use an adaptation strategy that merges different solutions. We also plan new solution if the system cannot find similar cases and always according to the user preferences.

The next stage of our research work will involve the work in the learning phase, after the solution has been successfully adapted to the new problem how to store the resulting experience as a new case in case base. For future work, we will attempt to ameliorate the web service composition by using agents like those presented in [33]. Finally we are looking for integrating the plan quality as an evaluation of the solution.

## REFERENCES

- [1] XML W3C. Extensible Markup Language (XML), <http://www.w3.org/XML/>.
- [2] SOAP Version 1.2 Part 0: Primer (Second Edition) W3C Recommendation. 27 April 2007.
- [3] WSDL W3C. Web Services Description Language (WSDL) Version 2.0 Part 1 : Core Language W3C Recommendation 26 June 2007.
- [4] UDDI Version 3.0.2. UDDI Spec Technical Committee Draft. <http://www.uddi.org/pubs/uddi-v3.0.2-0041019.htm>.
- [5] Hammond, K. J., Case-based planning: Viewing planning as a memory task . San Diego, CA: Academic Press,(1989).
- [6] Hu, J., Feng, Z., Automated Composition of Semantic Web Services Using Case-Based Planning. In :International Forum on Information Technology and Applications, (2009).
- [7] Liu, Y.Z., Qiu, S., Tao, H.J., Tian-Yizang, Wang, Y.D., A case-based reasoning approach to support web service composition. In: Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, (2009).
- [8] Aamodt, A. , Plaza, E. ,AI Com - Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59, (1994).
- [9] Hammond, K.J. Case-Based Planning: A Framework for Planning from Experience. *Cognitive Science* 14(3): pp. 385-443, (1990).
- [10] Limthanaphon, B., Zhang, Y., Web Service Composition with Case-Based Reasoning. In: Proceedings of the Fourteenth Australasian Database Conference, Adelaide, Australia pp.201–208, (2003).
- [11] Lajmi, S., Ghedira, C., Ghedira, K., WeSCo\_CBR: How to Compose Web services via Case Based Reasoning. In: Proceeding of the IEEE, International Conference on e-Business Engineering, Shanghai, pp.618-622 October (2006).
- [12] Thakker, D., Osman, T., Al-Dabass, D., Semantic-Driven Matchmaking and Composition of Web Services Using Case-Based Reasoning. In: Proceedings of the Fifth European Conference on Web Services, Washington, pp.67-76, November (2007).
- [13] Cheng, R., Su, S., Yang, F. and Li, Y. Using Case-Based Reasoning to Support Web Service Composition. In: ICCS 2006, Part IV, LNCS 3994, pp. 87 – 94, (2006).
- [14] Sun, Z., Han, J., Ma, D., A Unified CBR Approach for Web Services Discovery, Composition and Recommendation. In: International Conference on Machine Learning and Computing IPCSIT vol.3, IACSIT Press, Singapore, (2011).
- [15] Lee, C.L., Liu, A., and Huang, H., Using Planning and Case-Based Reasoning for Service Composition. In: J. Advanced Computational Intelligence and Intelligent Informatics, 14(5), pp. 540-548, (2010).
- [16] Massively parallel support for case based planning. Brian P. Kettler, James A. Hendler, William A. Anderson and Matthew P. Evett Department of computer science University of Maryland, College Park MD 20742. February 1994.
- [17] Lin, N., Web Service Composition with User Preferences. In: 5<sup>th</sup> European Semantic Web Conference, ESWC Tenerife, Spain, pp. 629-643, (2008).
- [18] Sohrabi, S. and Prokoshyna, N. and McIlraith, S.A., Web service composition via the customization of Golog programs with user preferences. In *Conceptual Modeling: Foundations and Applications*, pages 319--334, Springer-Verlag, (2009).
- [19] Bosca, A., Corno, F., Valetto, G., Maglione, R. On-the-fly construction of web services compositions from natural language requests, in: *Journal of Software* 1. pp. 40–50, (2006) .
- [20] Englmeier, K., Pereira, J., Mothe, J. Choreography of web services based on natural language storybooks. In: *Proceedings of the 8th International Conference on Electronic Commerce*, pp. 132–138, (2006).
- [21] Jang, M., Sohn, J.C., Cho, H.K., Automated question answering using semantic web services. In: *Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference*, pp. 344–348, (2007).
- [22] Gerevini, A., Long, D.: Plan constraints and preferences for PDDL3. Technical Report 2005-08-07, Department of Electronics for Automation, University of Brescia, Brescia, Italy (2005).
- [23] Gerevini, A., Long, D. Plan constraints and preferences in pddl3, ICAPS Workshop on Soft Constraints and Preferences in Planning, (2006).
- [24] A Heuristic Search Approach to Planning with Temporally Extended Preferences Jorge A. Baier and Fahiem Bacchus and Sheila A. McIlraith Department of Computer Science University of Toronto Toronto, Canada (2007).
- [25] Hsu, C.-W., Wah, B., Huang, R., and Chen, Y. Constraint partitioning for solving planning problems with trajectory constraints and goal preferences. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1924–1929, Hyderabad, India, (2007).
- [26] Lin, N., Kuter, U., Sirin, E.: Web service composition with user preferences. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 629–643. Springer, Heidelberg (2008).
- [27] Sohrabi, S., Baier, J.A., McIlraith, S.A.: HTN planning with preferences. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1790–1797, (2009).
- [28] Gerevini, A., Dimopoulos, Y., Haslum, P. and Saetti, A., Deterministic part, 5th International Planning Competition. <http://eracle.ing.unibs.it/ipc-5/>, 2006.
- [29] Mantaras, R.L.D., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. *Knowl. Eng. Rev.* 20(3), 215-240 (2005).
- [30] Munoz- Avila, H., Cox, M.: Case-based plan adaptation: An analysis and review. In: *IEEE Intelligent Systems* 23, 75-81, (2007).
- [31] Ram, A., Francis, A.: Multi-plan retrieval and adaptation in an experience-based agent. In: Leake, D.B. (ed.) *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, (1996).
- [32] Hanks, S., & Weld, D. A Domain-Independent Algorithm for Plan Adaptation. *Journal of Artificial Intelligence Research* 2 , pp. 319-360, (1995).
- [33] Al-Shargabi, B., Sabri, O., Sabri, A., Toward a User Preferences Agent Based Web Service Composition. *The International Journal of Information Technology & Computer Science (IJITCS)*, 14(2), 2014.

## Authors' Profiles

**Yamina Hachemi** is a PhD Student at the Computer Science Department of Sidi Bel Abbes University, Algeria. She received her Magister degree in Computer Science from Mascara University in 2010. An Assistant Professor since 2011



at Faculty of Economics, business studies and management of Mascara University, Algeria. She also received an engineer degree in computer science in 2004 from the Computer Science Department of USTO Oran University. Her researches interests include web service composition, semantic web; case based planning and user preferences.



**Sidi Mohamed Benslimane** is a Professor at the Computer Science Department of Sidi Bel Abbas University, Algeria. He received his PhD degree in computer science from Sidi Bel Abbas University in 2007. He also received a M.S. and a technical engineer degree in computer science in 2001 and 1994 respectively from the Computer Science Department of Sidi Bel Abbas University, Algeria. He is currently Head of Research Team 'Service Oriented Computing' at the Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS. His research interests include, semantic web, service oriented computing, ontology engineering, information and knowledge management, distributed and heterogeneous information systems and context-aware computing.