Asian Journal of Information Technology 4 (10): 957-961, 2005

Constraint Satisfaction in Dynamic Web Service Composition

Nizamuddin [1]Channa, [1]Li Shanping, [2]Shaikh Abdul Wasim and [1]Fu Xiangjun
[1]College of Computer Science, [2]Department of Mathematics, Zhejiang University, Hangzhou, China 310027

Abstract: Web services composition is a crucial aspect of web services technology, which gives us the opportunity for selecting new services and best suits our needs at the moment. Doing this automatically would require us to compute our criteria for composing candidate services.Present study represent all criteria guiding the selection of the services as constraints then choose an optimal set of services to satisfy the customers requirements. This approach reduces the dynamic composition of web services to a constraint satisfaction problem. This study will support in future to semantic web process.

Key words: Dyvamic web service, crucial aspect, optimal set

INTRODUCTION

In recent years, the growing numbers of Web Services (WSs) have emerged as the internet develops at a fast rate. The web is now evolving into a distributed device of computation from a collection of information resources[1]. The need for composing the existing WSs into more complex services is also increasing, mainly because of new and more useful solutions can be achieved. Hence, a collection of interacting WSs is more likely to accomplish these goals. Currently, users either utilize some services that they know already or find those services by looking it up in a keyword-based search engine (e.g. google) or by looking it up in a WSs registry like universal description, discovery and integration (UDDI)[2]. Present service composition research aims a treducing the complexity and time needed to generate and execute a composition and improving its efficiency by selecting the best possible services available at current time.

OVERVIEW OF DYNAMIC WEB SERVICES COMPOSITION AND CSP

Dynamic web services composition: Web services are modular, self-describing, self-contained applications that are accessible over the internet[3]. Web service composition[4] refers to the process of collaborating of the heterogeneous web services. It can be either static or dynamic. Indynamic composition, the web service to be used foran activity is decided at run-time. Dynamic composition involves run-time searching of registries to find services.

It is emerging as the technology of the choice for building cross-organizational applications on the web. The standards that are currently being considered for building process using web services composition include BPEL4WS[5] and OWL-S[6].

Constraints satisfaction problem: A constraints satisfaction problem (CSP)[7] is a problem to find a consistent assignment of values tothe variables. A CSP consists of a finite set of variables and a set of constraints. Each variable isassociated with a set of possible values, known as its domain. A constraint is a relation defined on some subset of these variables and denotes valid combinations of their values. A solution to a constraint satisfaction problem is an assignment of a value to each variable from its domain, such that all the constraints are satisfied. The Constraint Satisfaction Problem (CSP) model is widely used to represent and solve various Artificial Intelligence related problems such as scheduling, planning orcomputer aided design and provides fundamental tools in areas such as truth maintenance, expert systems or constraint logic programming. It also offers a simple, natural and yet powerful knowledge representation framework and various algorithms for solving various types of requests. In present study we have used constraint satisfaction approach as easily we find feasible set of candidate services.

MOTIVATION SCENARIO

Web service composition is the ability of one business to provide value-added services through composition of basic web services, possibly offered by different companies. The motivation of webservices composition is to create novel functionality by composing existing web services. It provides an end to

Corresponding Author: Nizamuddin Channa, College of Computer Science, Zhejiang University, Hangzhou, China 310027

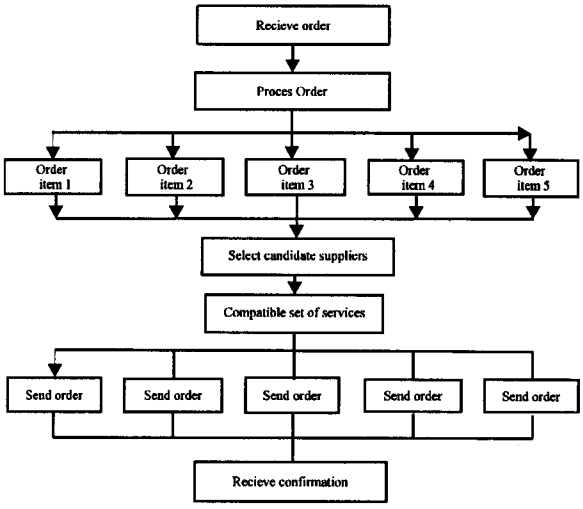

Fig. 1: Constraint based order process

end integration of business services with eachother in a dynamic manner. Web service composition provides a multiparty collaboration and interaction. And the developers are able to use a large number of web services provided by their parties to deliver more powerful and integrated business solutions

This presents a simple motivating scenario,where an automobile manufacture gets a large order for manufacturing auto mobiles and he must buy auto mobile parts from suppliers. The manufacturer would go to integrate a number of suppliers in their process and dynamically chose the suppliers based on quotations. In Figure 1, shows the logical steps of the overall flow. It starts by receiving the order from a customer. Then the order is processed and the potential suppliers are selected. This process also includes a step, where potential suppliers may be contacted for quotes. After getting the quotes, the best suppliers are chosen on the basis of constraintsand then orders are sent to them.

AN ARCHITECTURE OF DYNAMIC WEB SERVICE OMPOSITION

An architecture of dynamic web service composition represents the functionality of the required services using Service Templates (ST). The major components of the architecture are OWL-S/UDDI registries, discovery engine and constrain to ptimizer. Constraint optimizer constraint optimization discovery engine optimized service set service set cost estimation constraint representation OWL-S/ UDDI registries service requirements(Fig. 2).

OWL-S/UDDI registries: Service registry is a searchable registry of service description where service providers publish the irservice description and service consumer find service and obtain binding information for services during development for static binding or during execution for dynamic binding. OWL-S/UDDI registries decrease search time and increase the accuracy ofoperations like service discovery. The view adopted by OWL-S is that web services have a set of capabilities that they offer to the rest of the community. The capability of some Web services may be to provide information, as for instance stock quotes; the capability of other web service may be to provide services such as travel booking. In the irnormal operations, Web services solve problems through information exchange or the exchange of services.

The discovery engine: The discovery engine is an interface over OWL-S/UDDI registries to provide semantic publication and discovery. It also engages the use of metrics obtained by comparing the properties of the concepts, matching the cardinality and the data type, distance from the common parent, etc., in ranking the relevant services discovered. Discovery results returned by the user/tool are ranked according to the degree of match. In discovery engine, the discovery method is based primarily on the semantic descriptions and constraints advertised by the service provider. In discovery engine a query template is used to construct the query that specifies the function alaspects of the required service. The query template consists of specifics about a service such as operation name, operation action), input/output name and semantic type, exception, pre/post conditions, domain and location. Such a query may be generated by automated tools or built manually by users.

The constraint optimizer: To be able to select web service that are optimaland satisfy client's constraints requires the use of aconstraint optimizer. The constraint optimizer dynamically selects optimized services from candidate service set, which are returned by the discovery engine. The Constraint Optimizer module produces optimal service sets based on business technology and process constraints. The constraint optimizer makes sure that the discovered services satisfy the client request and then optimizes the service sets according to its requirement.

Constraint representation: The constraint representation module is a set of classes, attributes, instances and relationships between them that allows users to represent

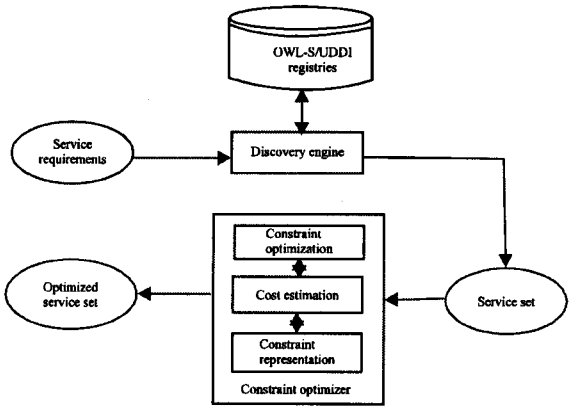

Fig. 2: An architecture of dynamic web service composition

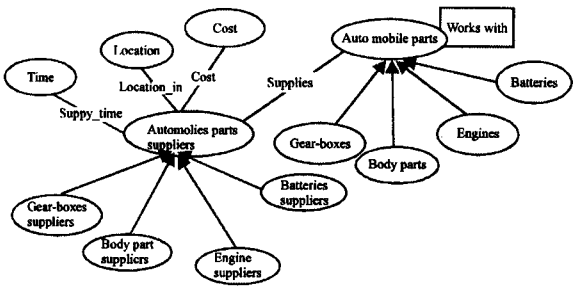

Fig. 3: Classes and their relationship in parts of automobile

the business constraints in ontologies. A business constraint may be any factor and rules that affects the selection of a web service for a given process. For example some suppliers may be preferred suppliers for one item but secondary suppliers for another item. There may exist a number of such business constraints for a particular process. For example, a secondary supplier may be chosen over a preferred supplier if it is cheaper. For illustration purposes, letus consider an example of representing business constraints. The automobile parts supplier ontology is depicted in Fig. 3, which represents the relationships between automobile part such as gear box, body parts, engine and battery. The ontology isused to capture the suppliers for each part, their relationships with the manufacturer and thetechnology constraints in their parts.

In Fig. 4 the ontology was developed in OWL[8] using the Protégé tool[9] and the instances in the ontology can be used to depict various relationships.

Let us examine some of the facts covered in the ontology. With the help of such instance statements the required constraints can be encoded in the format of

```
GEAR_BOXES_Supplier1 is an instance of
GEAR_BOXES_supplier , GEAR_BOXESSupplier1 supplies
#GEAR_BOXES_ Type1 and GEAR_BOXES_Supplier1 is a
preferred supplier.
<GEAR_BOXES_Supplier rdf:ID="GEAR_BOXES_Supplier1">
<supplies rdf:resource="#GEAR_BOXES_ Type1"/>
<supplierStatus>preferred</supplierStatus>
</GEAR_BOXES_Supplier>
GEAR_BOXES_ Type1 is an instance of GEAR_BOXES
GEAR_BOXES_ Type1 works with Battery_ Type1
<GEAR_BOXES rdf:ID="Type1GEAR_BOXES">
<worksWith>
<Battery rdf:ID="Type1Battery">
</worksWith>
</ GEAR_BOXES >
Battery_ _Supplier1 located in Hangzhou.
<Battery_ _Supplier rdf:ID=" Battery_ _Supplier ">
<Located in>
<Location rdf:ID="Hangzhou">
</Located in>
</Battery_ _Supplier>
```

Fig. 4: OWL expression of automobile parts

OWL and these constraints are important in deciding the suppliers.

Cost estimation: The cost estimation module queries theinformation stored in the constraint representation for estimating costs according to various factors, which affect the selection of the services for the dynamic composition processes.

Let us consider the supply chain for themanufacture. Here are some of the factors like cost, time, compatibility and reliability, which may affect the selection of the suppliers for a particular process. Depending on the manufacturers preferences at process execution, all the factors can be more or less important. For example, at a certain point of time a manufacturer may only want to deal with preferred suppliers, while at other times he may choose the lowest cost in order to be able to set priorities between these factors, the cost estimation module queries the constraint representation and assigns a cost value to certain process on the basis of theresults of the queries. The ontology is queried to getthe relationship status of the supplier using the queryin OWL-QL)[10](Fig. 5).

Here ss means supplier Status and s means supplier. If we query for GEAR_BOXES_Type1 based on the ontology, the preferred will bere turned.

The cost function for getting the value for this parameter is:

Supplier Status (ss) = 1, if ss is a preferred partner 1, supplier Status (ss) =0.5 if ss is a secondary supplier 2,

```
Query: ("What is the supplierStatus for a
GEAR_BOXES?") Query Pattern: {( supplierStatus ?ss ?s)
(type ?s GEAR_BOXES) }
Must-Bind Variables List: (?s)
```

Fig. 5: Query for supplier status

```
Query: ("type_x works type_y ?")
Query Pattern: {(works ?x ?y) (type ?x Type_x) (type ?y Type_y) }
Must-Bind Variables List: (?x,?y)
Answer Pattern: {(works ?x ?y)}
Answer: ("type_x works type_y")
Answer Pattern Instance: {( works "Type_x" "Type_y")}
```

Fig. 6: A composite service query

otherwise 0. Similarly for parts compatibility between the two suppliers, the ontology will be queried using the query Fig. 6. Compatibility (type_x, type_y) = 1, if part from type_x is compatible with part from type_y. 0, if part from type_x is not compatible with part from type_y 0.5, if type_x or type_y are not listed in the ontology.

Process Constraints: We refer to any constraints that apply to only that particular instance of the process as process constraints. For most metrics, the process QoS can be calculated using the aggregation operator's summation, multiplication, maximum or minimum. However, in some cases, the user has to define user-defined function for aggregation. In case of a number of scopes in the process, the constraints are written for each scope and process QoS is calculated by aggregation of all the scopes. In order to illustrate this, let us consider the following scenario for the manufacturer, where the manufacturer has to buy the parts and quantities (Table 1).

Let us assume that all these items are in the same scope. In that case, the process has only one scope.

```
Qos =< {},{},{},{'cost ','≤'80000 ','dollars','∑},
    {'supplytime ',' ≤ ',' 7 ','days ',' Max},
      {' partnerstatus ',' ≤ ',' Max'},
        {'compatibility ','≥',' 2 ','F1' },
          {'discovery score ','≥','Min'}>
```

Where F1 is the user-defined function, which calculates scope compatibility by using calling 'Compatibility ( ), ij ss ss ', (), ij .in scope ij ..

Constraint optimization: The constraints specified by the user are stored inservices template. The service providers can specify an operation in the service, which can be invoked toget the QoS metrics or constraints of the service. The optimization module retrieves constraints for the services matching the service template from either UDDI or by invoking an operation of the services pecified by the provider. These constraints and objective function when fed into the LINDO Integer Linear Programming solver, will produce an optimal set and a number of feasible sets, which would be ranked from optimal to nearoptimal solutions.

Table 1: Automobile part and their quantities

Part	Quantity
Body parts	3000
Gear boxes	3000
Engines	3000

RELATED WORK

Semantics has been proposed as key to increasingautomation in applying Web services and managing Web processes that take care of interactions between Web services to support business processes within and across enterprises. Academic approaches like WSMF and OWL-S have tried to approach this solution by using ontologies to describe Web services. This approach is consistent with the ideas of the semantic web, which tries to add greater meaningto all entities on the Web using ontologies. We believe that the ability to choose services dynamically is crucial to the success of the service-orientedarchitecture. An effort that comes closest to present study is quality driven web services composition[11], they proposed a linear programming approach to optimize service selection across the process using generic QoS parameters. While they focus solely on optimization on generic QoS issues, we provide a comprehensive frame work, which optimizes service selection based on multidimensional criteria such as cost estimation, process constraint and Qos.

CONCLUSIONS

In this study, we have presented an approach forconstraint satisfaction in dynamic Web service composition. In this study constraint optimizer module uses an integer linear programming solver for process optimization based on process and business constraints. We have use the cost estimation module which queries the information stored in the constraint representation module for estimating costs for various factors, which affect the selection of the services for the processes. In the future, we plan to consider validation of the composition based on OWL-S standards like Input, output Pre and post-condition composition methodology, which can help user in an interactive composition.

REFERENCES

1. Curbera, F., W. Nagy and S. Weerawarana, 2001, Web Services: Why and how. Workshopon object-oriented web services. OOPSLA Tampa, Florida, USA.

2. Universal description, discovery and Integration of Web Services,http://www.uddi.org/
3. Fensel, D. and C. Bussler, 2002. Semantic webenabled web services. In Proceedings of International Semantic Web Conference (ISWC'2002), volume 2342.
4. Benatallah B., M. Dumas, M. C. Fauvet and F.A. Rabhi, 2002. Towards Patterns ofWeb Services Composition. In Gorlatch, S. and F. Rabhi (Eds.), Patterns and Skeletons for Parallel and Distributed Computing. Springer Verlag (UK). Business Process Execution Language for Web Services, version
5. ftp://www6.software.ibm.com/software/developer/library/ws-bpel11.pdf
6. OWL-S: Semantic Markup for web services. OWL-white paperhttp://www.daml.org/services/owl-s/1.0/owl-s.pdf.
7. Kumar, V., 1992. Algorithms for Constraint-Satisfaction problems: A survey. J. AI Magazine, 13: 32.
8. http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1sdfsd
9. [http://protege.stanford.edu/download/prerelease/full/
10. Fikes, R. and H. Patrick, H. Ian., OWL-QL. A language for deductivequery answering on the semantic web. J. Web Semantics.
11. Zeng, L., B. Benatallah, M. Dumas, J.Kalagnanam and Q. Sheng, 2003. Quality driven webservices composition. May 20–24, Budapest, Hungary. ACM.