AUTOMATIC WEB SERVICE SELECTION BY OPTIMIZING COST OF COMPOSITION IN SLAKY COMPOSER USING ASSIGNMENT MINIMIZATION APPROACH

P. Sandhya
Sathyabama University, Research Scholar, Department of Information Technology,
Jeppiaar Nagar, Rajiv Gandhi Road, Chennai - 600 119. Tamilnadu, INDIA.,
catherinesandhya@gmail.com

Dr. M. Lakshmi
Sathyabama University, Head, Department of Computer Science and Engineering,
Jeppiaar Nagar, Rajiv Gandhi Road, Chennai - 600 119. Tamilnadu, INDIA.,
laks@icads.co.in

Abstract

Web service composition is a means of building enterprises virtually by knitting relevant web services on the fly. Automatic web service composition is done dynamically at runtime. Extensive research has been done in the field of automatic web service composition. However all the works focus on providing client oriented results and hence there is less industry adoption of composition technology. In this paper we have proposed a new service collaboration stack that composes with realistic business metrics of a provider in addition to client metrics. Some of the service provider metrics include time planning, profit management, native intelligence, user adoption, environment, market scenario, vision and industry adoption. In this paper we focus on enhancing industry adoption through optimizing cost of service composition. We propose the SLAKY composer that solves assignment of appropriate service during composition as an assignment minimization problem to reduce the cost of composition. We also extend OWL-S profile sub ontology to augment cost as a service parameter.

Keywords: SLAKY Composer; Automatic web service composition; Assignment minimization problem, OWL-S Ontology.

1. Introduction

Web service as defined by [Haas and Brown (2010)] is a software system designed to support interoperable machine-to-machine interaction over the network. It has an interface in machine processable format WSDL. Other systems interact with the web service in a manner prescribed by its description using SOAP-messages, conveyed using HTTP with an XML Serialization and other related standards. [Broberg(2002)] defines web services as a software component that is described via WSDL and is capable of being accessed via standard network protocols such as SOAP over HTTP. Traditional business models have now transformed to virtual business models. Web service composition is aggregation of simple services creating virtual enterprises on the fly. There are several types of web service compositions. Static composition composes services at design time. Dynamic and automatic compositions are done at runtime. Dynamic composition deals with non deterministic nature of services. On the other hand automatic web service composition deals with the ease of composition than done manually. Static web service composition can be designed using workflow languages. For automation of service composition by an agent the services have to be annotated. Annotations are done using OWL-S or WSMO upper ontologies. Business models based automatic web service composition is at its infancy. Deciding on a business collaboration partner service on the fly is rather unrealistic. Though several works have been done on web service composition; much of the work deals with providing appropriate service for the client. The business specific metrics of the service provider is not captured and hence lacks industry adoption. In this paper we capture provider specific metrics and collaborate with other services in a realistic manner. When business is driven through service composition; industry adoption will be done if the cost of composition is at the minimum. In this paper we semantically annotate client services with their cost of composition with other services. We hence arrive at an assignment minimization problem. We extend service profile sub ontology to augment cost of composition as a service parameter.

2. Related Works

There are several works on automating web service composition. [O.Hioual and Z.Boufaida(2011)] proposed semantic web service composition in an eb-XML environment. [K. Sycara, M. Paolucci, A. Ankolekar and N. Srinivasan (2003)] has proposed web service composition based on DAML-S. [M.Pasha and H. Farouq Ahmad (2008)] developed negotiation between agent and semantic web services. [Maximilien, E., M., Singh, M., P., (2003)] propose an approach wherein middle agents serve as proxies for Web services to assist an application in selecting implementations that best match the quality criteria of the application. [. Pistore, F. Barbon, P. Bertoli, D. Shaparau & P. Traverso (2004)] proposed semantic web service composition based on planning as model checking. [Narayanan, S., and McIlraith, S.(2002)] encoded service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. [Sirin, Hendler, (2003)] presented matching services to the user at each step of a composition, filtering the possibilities by using semantic descriptions of the services. [Arpinar , Ruoyan Zhang , Boanerges Aleman-meza , Angela Maduko, (2004)] proposed Interface-Matching Automatic composition technique. [Jorge Cardoso , Amit Sheth (2003)] encapsulated an organization's functionality within an appropriate interface and advertise it as Web services. [Kaarthik Sivashanmugam , Kunal Verma , Amit Sheth , John Miller (2003)] added semantics to WSDL using DAML+OIL ontologies. [G. C. Gannod, R. J. Brodie and J. T. E Timm, (2005)] proposed an approach for generating groundings for a semantic Web service and demonstrated how the use of lightweight interactive tools facilitates creation of groundings for a semantic Web service. We find that most of the above stated composition techniques focus on client retention or ease of programmer to compose. In this paper we augment realistic service provider metrics which otherwise would become a toy model.

3. Assignment Minimization Problem

Assignment problem involves allocation of n different facilities to n different task such that the cost is minimal [wiki]. For example a company has n person with different capabilities performing each a job and there are same number of jobs but of different types. Each job can be assigned to one person only. Let Cij be the cost of assigning a person i to a job j. The assignment problem will determine which job is to be assigned to which person such that the cost is minimal. We formulate this in linear programming as follows:

$$\text{Xij} = \begin{cases} 1 \; if \; job \; j \; is \; assigned \; to \; person \; i \\ 0 \; if \; job \; j \; is \; not \; assigned \; to \; person \; i \end{cases}$$

As each person is assigned exactly one job we frame the following:

$$\sum_{j=1}^{n} Xij \; for \; i = 1,2,3..n$$

Similarly each job is assigned to one person:

$$\sum_{i=1}^{n} Xij \; for \; j = 1,2,3..n$$

The objective of assignment problem is to minimize Z =

$$\sum_{i=1}^{n} \sum_{j=1}^{n} Cij \; Xij$$

In this paper we promote industry adoption of service oriented business model by effective planning of cost of composition. Considering n services of a service category A to be composed with n services of another category B such that there is an one to one composition with minimal compositional cost. Thus automatic composition can be coined as an Assignment minimization problem.

4. Need for Optimization of Composition Cost

Consider that an agent has n services of functional type A and B. The cost of composition is the aggregated cost of each process in the composition. Set A services need to be assigned with set B services such that the compositional cost is optimal. We resolve this issue by assignment minimization approach. The cost matrix contains cost of all possible compositions. The cost of process composition is augmented in OWL-S service profile sub ontology. OWL-S is the upper ontology for web service composition. The assignment problem will

determine which job is to be assigned to which person such that the cost is minimal. We formulate this in linear programming as follows:

$$Xij = \begin{cases} 1 \; if \; B \; Service \; j \; is \; assigned \; to \; A \; Service \; i \\ 0 \; if \; B \; Service \; j \; is \; not \; assigned \; to \; A \; Service \; i \end{cases}$$

As each Service of type A is assigned exactly one Service of type B, we frame the following:

$$\sum_{j=1}^{n} Xij \; for \; i = 1,2,3..n$$

Similarly each Service of type B is assigned exactly one Service of type A

$$\sum_{i=1}^{n} Xij \; for \; j = 1,2,3..n$$

The objective of assignment problem is to minimize Z(Cost of composition) =

$$\sum_{i=1}^{n} \sum_{j=1}^{n} Cij \; Xij$$

5. OWL-S Upper Ontology

WSDL describes services at syntactic level requiring human intervention for composition of services [Grigoris Antoniou, Frank van Harmelen (2003)]. To automate composition meaningfully the semantics of service functionality have to be described using ontology. To obtain service descriptions there are two kinds of ontologies namely generic web service ontology like OWL-S to specify input, output, precondition and effect and a domain ontology to specify web service domain knowledge such as service parameters and domain ontology. The generic web service ontology has four subontologies namely Profile, Process, Grounding and Service. Profile ontology specifies what a service does, Process ontology specifies how the service works and Grounding ontology specifies how the service is implemented. The Service ontology links ServiceProfile, ServiceProcess and ServiceGrounding which are further specialized as sub-concepts in Profile, Process and Grounding ontology.The profile ontology specifies what the service does as a functionality offered by the service, semantic type of inputs outputs precondition effect, details of service provider, and several service parameters like quality rating and geographic radius. Service discovery is based on this description. For each of the functionality of a specific service there is a profile. The hasProc relation specifies the association of each Profile instance with the process it describes.Process ontology describes the internal process models of complex services. The process model is used to automatically compose the web services. The Process ontology has a concept called ProcessModel which describes a single Process. The process can be atomic, simple or composite with or without control constructs. The IOPE of Profile and Process are linked by refersTo property. Grounding ontology describes the vocabulary to link conceptual description of service specified by Profile and Process to actual implementation details. The implementation details include network protocols, message etc. The WSDLGrounding contains WSDL description that treats AtomicProcess as a WSDL operation. Therefore the inputs and outputs of AtomicProcess are mapped to corresponding messagepart in the input message and output message respectively of WSDL operation. The Grounding ontology specializes ServiceGrounding as a WSDLGrounding. The type of WSDL messagepart is specified as a OWL-S parameter. Each WsdlAtomicProcessGrounding elements of WSDLGrounding grounds an atomic process in ProcessModel.The semantic description of service is offered by Profile and Process ontology. WSDL offers the syntactic descriptions of the service. The mapping between syntactic and semantic descriptions is given by Grounding ontology.To enrich OWL-S ontology with domain knowledge domain descriptions are specified in domain ontology. The domain ontology is based on OWL and contains a DataStructure and Functionality hierarchy. The functionality hierarchy supports better discovery than a normal key-word based discovery. We extend OWL-S upper ontology to include cost specification as a service parameter in service profile ontology.

6. SLAKY Collaboration Stack

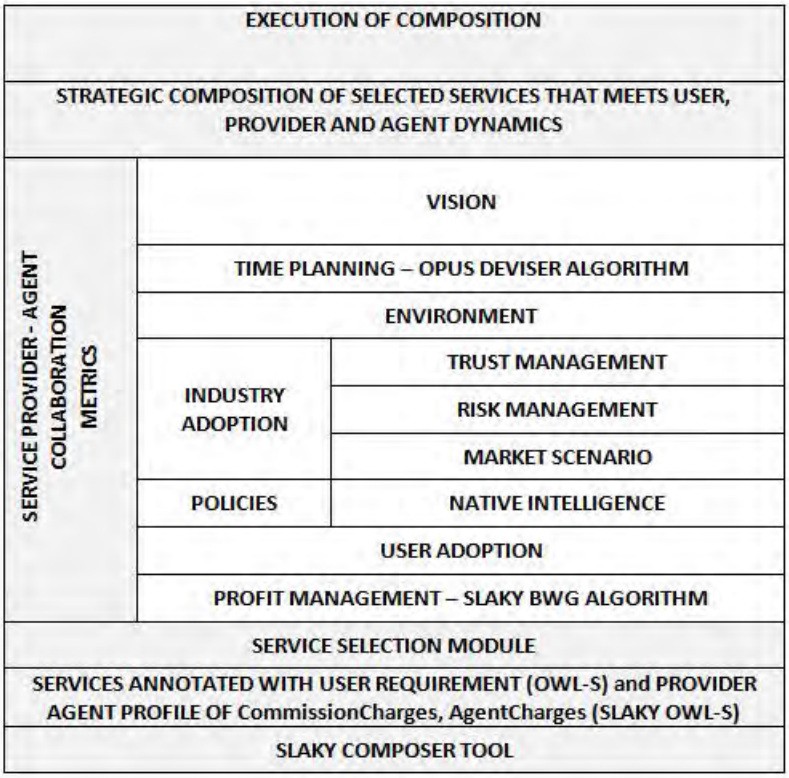

Fig. 1. SLAKY Collaboration Stack

SLAKY is a realistic model for choosing business service partners considering service partner collaboration metrics including vision, time planning, environmental context, user adoption, usage policies, trust management, risk management, market scenario, native intelligence and competitive profit management of service partners apart from functionality satisfaction for client's requirements. We have proposed SLAKY architecture because the choice of a service partner should not only satisfy end user's requirements but also meet real world business objectives of the composite application provider. SLAKY composer chooses services such that they give high profit computed by profit management module using SLAKY BWG algorithm [P. Sandhya, M. Lakshmi (2011)]. The time planning module was designed using opus deviser algorithm [P. Sandhya, M. Lakshmi (2012)]. The native intelligence module was designed using Naavi algorithm [P. Sandhya, M. Lakshmi (2012)]. In this paper to promote industry adoption in a realistic manner compositions must be chosen strategically such that composition cost is minimized. As discussed earlier we use assignment minimization approach for strategic composition.

7. Industry Adoption Enhancement Module Architecture

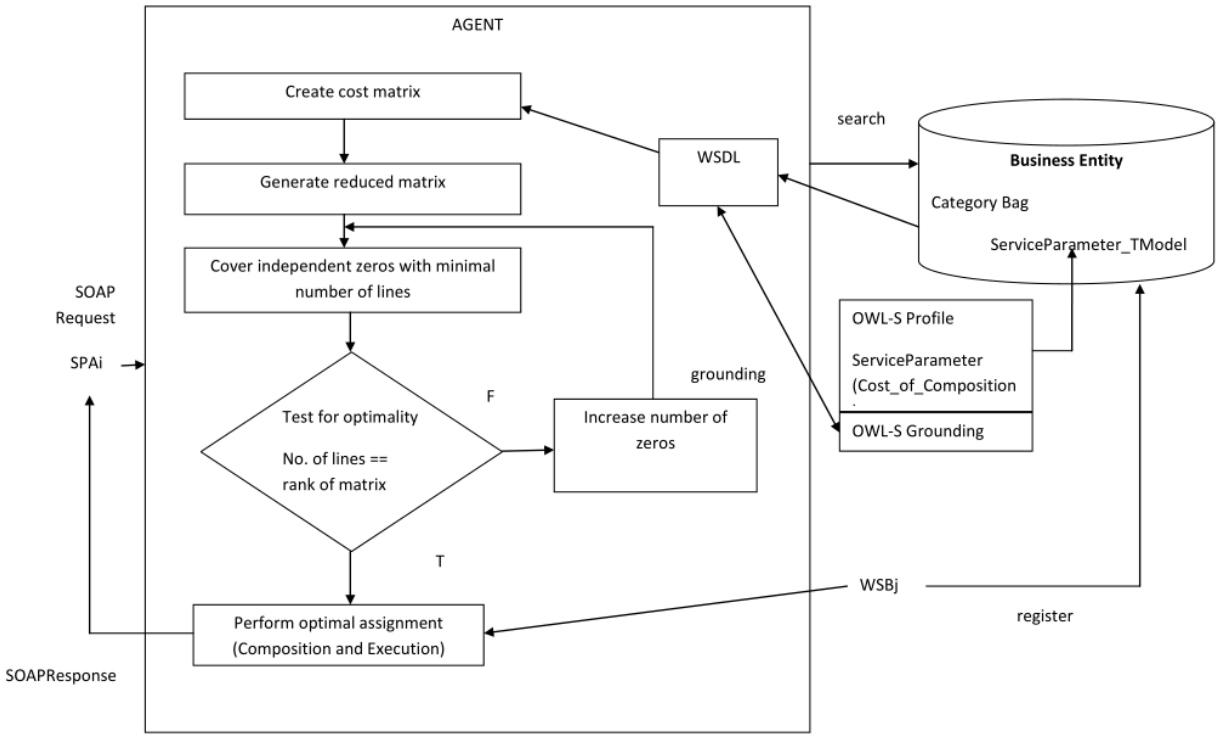

Fig. 2. System Architecture

The cost of composition is the service execution charges. Service execution charges are the summation of hardware charges h, storage charges, power charges p and miscellaneous charges m (tax, discounts, financial policies, etc..).

Service execution charges c = h + s + p + (or) - m;

Services themselves can be composite. For instance if a composite service is an aggregation of n simple services then

$$c = \sum_{i=1}^{n} hi + si + pi + (or) - mi$$

When services partners collaborate the compositional cost must be optimal. For instance when a manufacturer requests raw materials from suppliers there may be n number of suppliers. It is a tendency for the manufacturer to purchase from the supplier the raw materials at optimal cost without compromising the quality of service. Thus corporate adoption of services is mathematically solved as an assignment minimization problem.

Let us consider that there are n services of type A need to consume n services of type B. Each type B service has annotated their bids. Each service of type A composes with any one service of type B. The composer agent has to assign (compose) appropriate service of type B to A such that the cost of composition is minimal.

Steps for assignment minimization oriented composition

Let C be an n X n cost matrix.

Step 1: Modify cost matrix to get atleast one zero in each row and column. To do so, subtract the smallest element of each row from other elements of that row.

Step 2: Subtract smallest element in each column from rest of the elements of the same column.

Step 3: Draw the minimal number of horizontal or vertical lines so as to cover all zeros.

Step 4: Perform test for optimality. Check the number of lines is equal to the rank of the matrix. If not increase the number of zeros till numbered lines becomes equal to the rank of the matrix. To do so; select the smallest element that does not have a line through it and subtract it from all uncovered elements. Add this smallest number to all elements at intersection of lines.

Step 5: Test for optimality again.

Step 6: To make optimal assignment mark all zeros, wherever there is a single zero in a row or column and crossout the rest of the zeros appearing in corresponding columns and rows. The marked zeros are the

assignments (compositions) to be made. The crossed zeros indicate that the relevant row or column is no longer available for further composition.

ALGORITHM:

Flood's Method

```
Flood (C: n×n matrix of cost)
        for  i :=1 to n
                 begin
                          u_i:= smallest cost in row i of C
                         for  j := 1 to n
                          c_ijnew:= c_ij− u_i
                end
        for j := 1 to n
                begin
                          v_j:= smallest integer in column j of C_new
                          for i := 1 to n
                          c_ijnew:=  c_ijnew− v_j
                end
{C_new is now the reduced matrix}
Z := an independent set of zeros of maximal size in C_new
q := |Z|
        while q < n //n-rank of matrix
                begin
                          cover(C_new)
                          k := smallest entry in C_new not covered by a line
                          for i := 1 to n
                          for j := 1 to n
                          begin
                                    if c_ijnew is not covered then c_ijnew:= c_ijnew − k
                                    if c_ijnew is covered twice then c_ijnew := c_ijnew + k
                          end
                Z := an independent set of zeros of maximal size in c_new
                q :=|Z|
                end
 for i := 1 to n
for j := 1 to n
if c_ijnew ∈ Z then
x_*ij:= 1
else
x_*ij:= 0
{X*= [x_*ij] is an optimal solution}
```

8. Extended OWL-S Ontology

The agent decides to assign service composition with minimal charges. Therefore the services have to augment their cost of composition as a service parameter in profile ontology of OWL-S upper ontology.

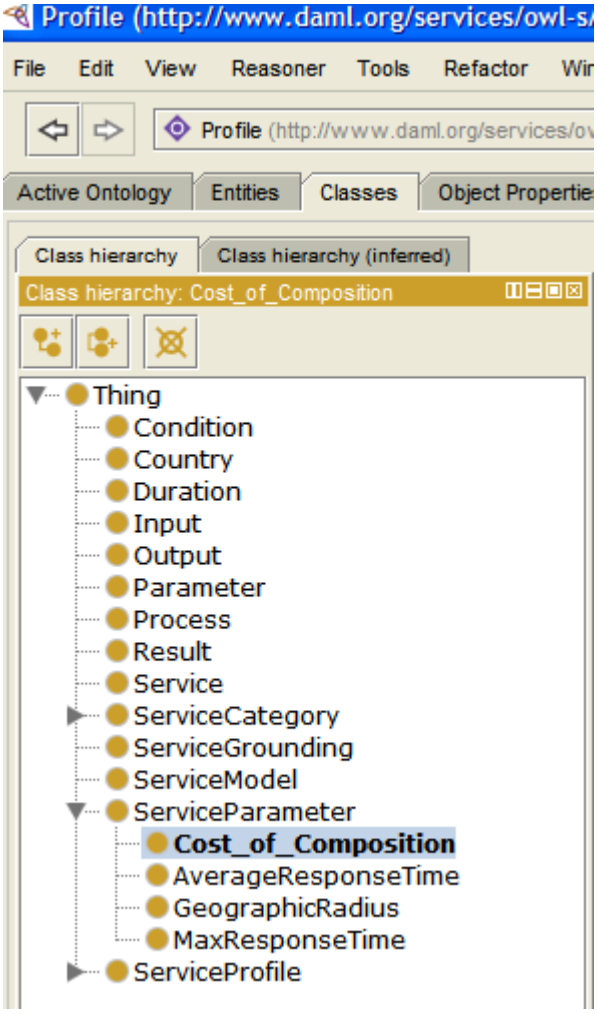

Fig. 3. Extended OWL-S

The agent uses assignment minimization to minimize the cost. Every service thus has to publish the cost of composition as specified by the extended OWL-S service parameter.

9. Implementation

In this paper we consider four service providers providing service A (SPAi where i=4) and four service providers providing service B (SPBj where j=4). The cost of composing services of type A and B is populated as a matrix. The cost matrix thus has rank four

n (Number of services of each type) = 4 and rank of cost matrix = 4

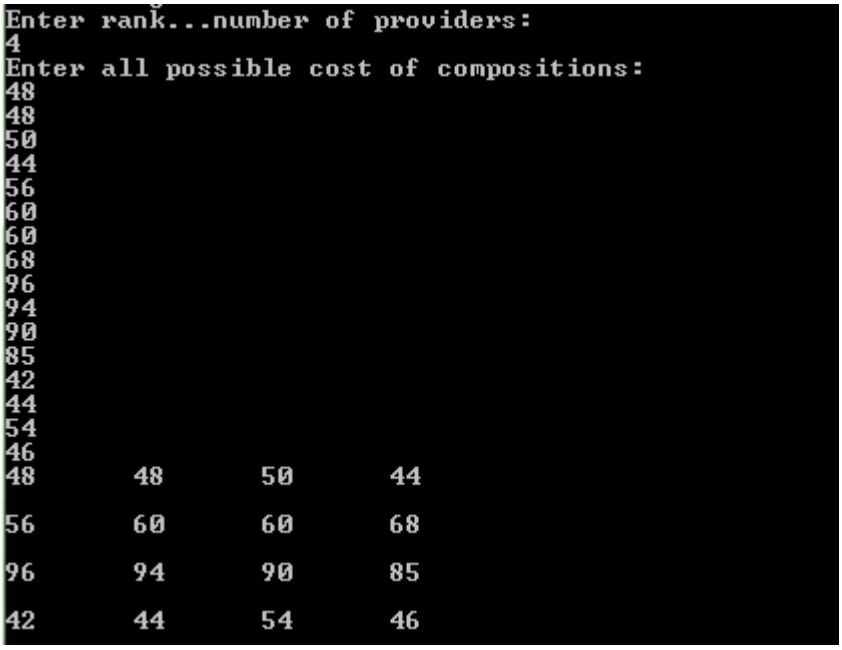

Fig. 4. Cost Matrix

The goal is to determine which SPBj service is to be composed with SPAi service keeping composition cost minimally. The cost matrix is reduced by subtracting smallest element of each row from each element of the same row. The cost matrix is further reduced by subtracting smallest element of each column from each element of the same column. The reduced matrix is given below.

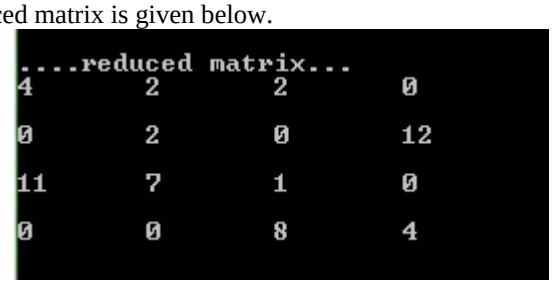

Fig. 5. Reduced Matrix

Minimum number of horizontal and vertical lines is drawn to cover all zeros. In this case

Horizontal lines are – row 2 and row 4

Vertical lines are - column 4

Now we perform the test for optimality. An optimal solution is reached if number of covered lines (q) is equal to rank of matrix. The optimality test fails because q=3 and rank=4

Iteration towards an optimal solution must be performed. To do this select smallest element that is uncovered. Then subtract it from all other elements that are not covered and add to all intersections of coverings.

Draw minimum number of horizontal and verticals lines to cover all zeros.

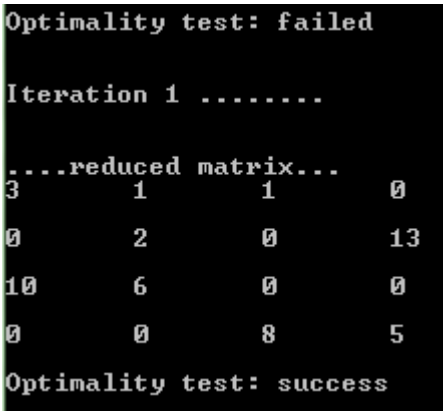

Fig. 6. Iteration 1: Reduced Matrix

Horizontal lines are – row2, row3, row 4
Vertical lines are – Column 4
The test for optimality succeeds as rank and number of covered lines (q=4) are equal.
Thus the following compositions are optimal.

Fig. 7. Optimized compositions

We have tested the n X n services without assignment minimization. The agent selects services based on user metrics alone. Then the cost of composition is 244 which is not optimized.

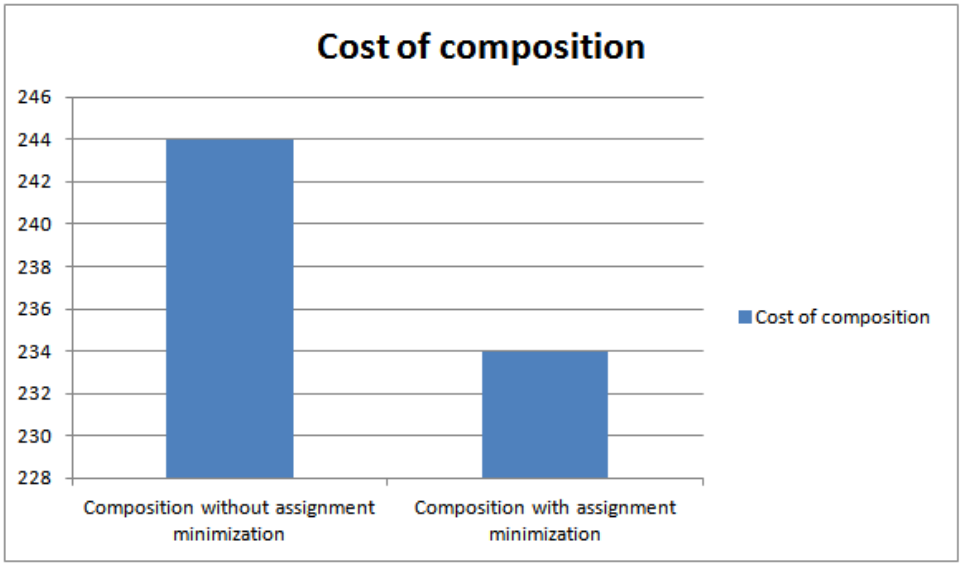

Fig. 8. Comparison of composition with and without assignment minimization

10. Conclusion and Future Enhancement

For industries to really adopt composite business model business oriented metrics have to be integrated. All the works on web service composition provides good service to clients but is deficient of business metrics of service provider. In this paper we have introduced industry adoption as a service provider metric. Providers would collaborate and adopt with other services such that cost of composition is minimal. We augment Cost_of_Composition metrics as a service parameter in OWL-S profile sub ontology. To assign appropriate services for composition with minimal cost we use the assignment minimization approach. The graph shows that composition with assignment minimization offers optimal cost of composition. Minimizing cost of composition promotes industry adoption. In this paper we have considered the number of service providers of type A and B to be equal. The resulting cost matrix is a square matrix. However in real world scenario the number of service

providers of type A and B may vary resulting in a non-square matrix. As a future enhancement in such a case the proposed algorithm can add dummies to the non-square matrix.

References

[1] Haas and Brown, (2010), Semantic web – Concepts, Tecgnologies and Applications, pg.127, Springer International Edition

[2] Broberg J (2002), Semantic web – Concepts, Tecgnologies and Applications, pg.128, Springer International Edition

[3] O.Hioual and Z.Boufaida (2011), An Agent Based Architecture (Usingplanning) For Dynamic And Semantic Web Services Composition In An Ebxml Context, International Journal of Database Management Systems (IJDMS), Vol.3, No.1, February 2011, pg. 111-131

[4] K.Sycara, M.Paolucci, A. Ankolekar & N. Srinivasan (2003) "Automated discovery, interaction and composition of semantic web services", Journal of Web Semantics 1 (2003), pp 27-46

[5] M.Pasha and H. Farouq Ahmad (2008) ,Agents Negotiating with Semantic Web Services, Proceedings of the World Congress on Engineering and Computer Science 2008, WCECS 2008, 22 - 24, 2008, ISBN: 978-988-98671-0-2, San Francisco, USA

[6] Maximilien, E., M., Singh, M., P., July(2003), Agent-based architecture for autonomic web service selection, In Proceedings. of the 1st International Workshop on Web Services and Agent Based Engineering, Sydney, Australia

[7] P. Traverso (M. Pistore, F. Barbon, P. Bertoli, D. Shaparau & 2004), Planning and MonitoringWeb Service Composition, Proceedings of the International Conference on Artificial Intelligence,Methodology,Systems, and Applications, p. 106-115.

[8] Narayanan, S., and McIlraith, S.(2002). Simulation, Verification and Automated Composition of Web Services, InProceedings of the Eleventh International World Wide Web Conference (WWW-11).

[9] Sirin, Hendler, (2003), Semi-automatic Composition of Web Services using Semantic descriptions. 1st Workshop on Web Services: Modeling, Architecture and Infrastructure in conjunction with ICEIS.

[10] Budak Arpinar , Ruoyan Zhang , Boanerges Aleman-meza , Angela Maduko, (2004), Ontology-driven Web services composition platform

[11] Jorge Cardoso , Amit Sheth (2003), Semantic E-Workflow Composition, Journal of Intelligent Information Systems

[12] Kaarthik Sivashanmugam , Kunal Verma , Amit Sheth , John Miller (2003), Adding Semantics to Web Services Standards

[13] Bacem Wali, Bernard, (2003), Gibaud, Extending OWL-S for the Composition of Web Services Generated With a Legacy Application Wrapper

[14] G. C. Gannod, R. J. Brodie and J. T. E Timm, (2005) An Interactive Approach for Specifying OWL-S Groundings, EDOC, 2005, pp. 251-260.

[15] http://en.wikipedia.org/wiki/Assignment_problem

[16] Grigoris Antoniou, Frank van Harmelen (2003), A Semantic Web Primer, The MIT Press, Cambridge, Massachusetts, London, England

[17] P. Sandhya, Dr. M. Lakshmi, (2011), A Novel Approach for Realizing Business Agility through Temporally Planned Automatic Web Service Composition using Network Analysis, http://www.lidi.info.unlp.edu.ar/worldcomp2011-mirror/sww2789.pdf, SWWS 2011, Lasvegas, Neveda.

[18] P. Sandhya, Dr. M. Lakshmi, (2012), Strategic Composition of Semantic Web Services using SLAKY Composer, WEST2012, Chennai, India, Springer Proceedings, Springer-Verlag Berlin, Heidelberg ©2012, http://link.springer.com/chapter/10.1007%2F978-3-642-31600-5_40?LI=true.

[19] P. Sandhya, Dr. M. Lakshmi, (2012), Modeling Native Intelligence Semantics For Indigenous Election Of Web Services Using Slaky Composer, International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.2, No.5, October 2012, pg. 15-30.