Multi-Agent Based Semantic Web Service Composition Models

SANDEEP KUMAR[1]
R. B. MISHRA
Department of Computer Engineering, Institute of Technology,
Banaras Hindu University, (IT-BHU), Varanasi-221005, India
[1](sandeepkumargarg@gmail.com, sandeep.garg.cse06@itbhu.ac.in)

Abstract. The semantic web service composition systems can also be implemented using the concept of multi-agent systems. In this approach, each of the agent capability is used to serve a particular service request. Paper presents two models for multi-agent based semantic web service composition process, differing on the point of using a coordinator for controlling the process or not. Further, based on one of the model, a novel multi-agent based semantic web service composition approach has been presented, which deals with some of the important issues in composition process. The approach presented is further applied by exploring a new area i.e. education planning as an application of semantic web technology.

Keywords: agent, application, composition, semantic web.

(Received March 12, 2008 / Accepted May 26, 2008)

1 Introduction

Semantic Web Services (SWSs) have modular structure and can be published, located or called (invoked) through the web. The different services can be combined with other homogeneous or heterogeneous services to form complex web applications. So, the interfaces, properties, capabilities, and effects of SWSs are encoded in a machine-understandable form to allow an easy integration of heterogeneous services. This process of generating aggregated service by the integration of independent available component services for satisfying a client request that can not be satisfied by any single available service is called as Semantic Web Service Composition.

A Multi-Agent System (MAS) consists of a team or organization of software agents, collectively performing a task which could not be performed by any individual agent. This paper is based on the understanding that a SWS Composition system can be considered as a MAS, where each component service is considered as an agent capability implemented as a self-contained software component.

The paper presents two possible models for MAS based SWS composition process. The work also presents a novel MAS based SWS composition approach based on one of the presented model. A similar work has been presented by the work in [3]. However, except [3], no other work using similar understanding was found. [3] have presented a framework for agent-enabled SWS composition. But some of the issues in composition process are there, which are not discussed till now. We have presented a detailed overview of these issues and have handled in our proposed approach.

The paper has been structured as follow. Apart from introduction in section-1, section-2 presents two MAS based SWS composition models. Section-3 presents a novel multi-agent based SWS composition approach and the section-4 presents some implementation issues regarding the presented approach with exploring a new application area i.e. education planning. The paper has been concluded in section-5.

2 MAS Based SWS Composition Models

Depending on the variations in the composition process, we have presented two models for MAS based SWS composition process. These models basically vary on the use of coordinator agents in the composition process. Figure 1 shows model for composition process, in which no coordinator agent is used. In this model, if required, the input request from User Agent U is directly decomposed by the system into atomic task/activities $Task_{1,}$ $Task2_{,}$ $Task_{3}$ ... $Task_{n,}$ based on its ontological description. After that, for each of the atomic task, the candidate software agents, who are acting as semantic web service components, are discovered and finally the best one is selected using agent-selection parameters. $SPA_{1,}$ $SPA_{2,}$ SPA_{3} ... SPA_{n} are the selected service provider agents (SPA) for tasks $Task_{1,}$ $Task2_{,}$ $Task_{3}$ ... $Task_{n}$ respectively. The user agent U, now negotiate with each of the SPA and assign it the respective task. The arrangement for the negotiation can take place using FIPA Contract Net Protocol [12] and SPA can accept task by means of agent's communication interface built upon FIPA-ACL [4]. Service is invoked by the SPA via the interface specified by its binding description.

Figures 2(a) and 2(b) shows the composition model for which a coordinator agent is used which control the complete MAS based SWS composition process. This model further has two variations of using an independent dedicated coordinator agent and using a coordinator agent which in addition to coordinating the composition process also perform some of the atomic tasks, as shown by Figures 2(a) and 2(b) respectively. In the first variation, as shown in Figures 2(a), the user agent U gives input request into the composition system, which is then specified in the terms of ontology. Using the parameters specified in the ontology description of request, the candidate coordinator agents are discovered and finally a best one is selected. The selected coordinator agent C now decides from ontology description, if the input request is atomic activity or it is complex one. In case, the request is complex one, it is decomposed into atomic tasks $Task_{1,}$ $Task2_{,}$ $Task_{3}$ ... $Task_{n}$. However, before decomposing the request, C can perform a validation over the input request to check if all the parameters, preferences and constraints specified in the request are proper or not. Now, Coordinator agent C discover and then select the corresponding service provider agents $SPA_{1,}$ $SPA_{2,}$ SPA_{3} ... SPA_{n} for atomic tasks $Task_{1,}$ $Task2_{,}$ $Task_{3}$ ... $Task_{n}$ respectively.

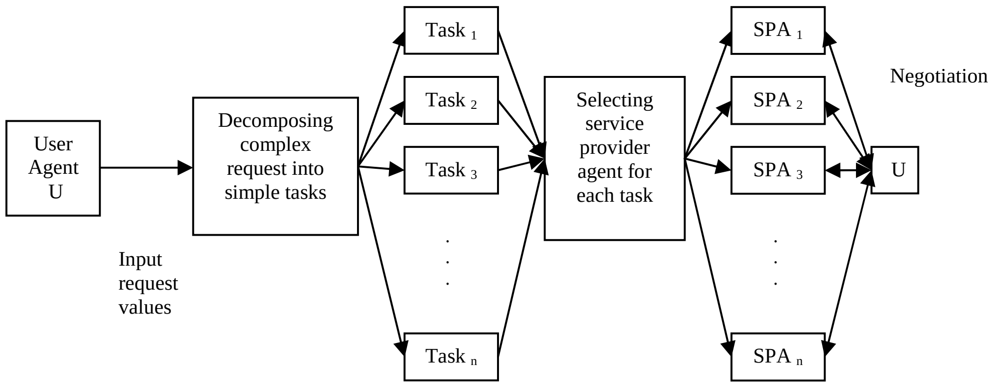

Figure 1: SWS Composition System without coordinator agent

The coordinator agent C negotiate with the SPAs and assign tasks, which further invoke the services, in the similar way described for the first model. The second variation, as shown in Figure 2(b), works similar to the first one except that in this model, the coordinator agent C can perform some of the tasks, in addition to coordinating the composition process. So, C only select the service provider agents $SPA_{c1,}$ $SPA_{c2,}$ SPA_{c3} ... SPA_{ck} for the tasks $Task_{c1,}$ $Task_{c2,}$ $Task_{c3}$ … $Task_{ck,}$ from among the total decomposed tasks $Task_{1,}$ $Task2_{,}$ $Task_{3}$ ... $Task_{n}$ for which it is not capable to perform on its own.

The model with coordinator is advantageous than the other one due to the reason that in this model, some of the constraints and parameters put by the user are validated by the coordinator agent C, before performing the actual composition process and hence saving the time and resources. For example, the customer may put some constraints like constraints over the budget of his/her request, constraints over the time in which the request has to be satisfied, and constraints over the expenditure on service fees of SPAs. This is the possibility that the constraints are such that within these constraints the satisfaction of request seems near to impossible. On the other hand, all these types of constraints in the request can not be checked by any one of the SPAs. So, in the first model, after the negotiation process with all the SPAs are over, then only it is decidable that the request can be satisfied within the constraints or not and if any one of the constraints is not satisfiable even after the negotiation then that flow of composition process will fail.

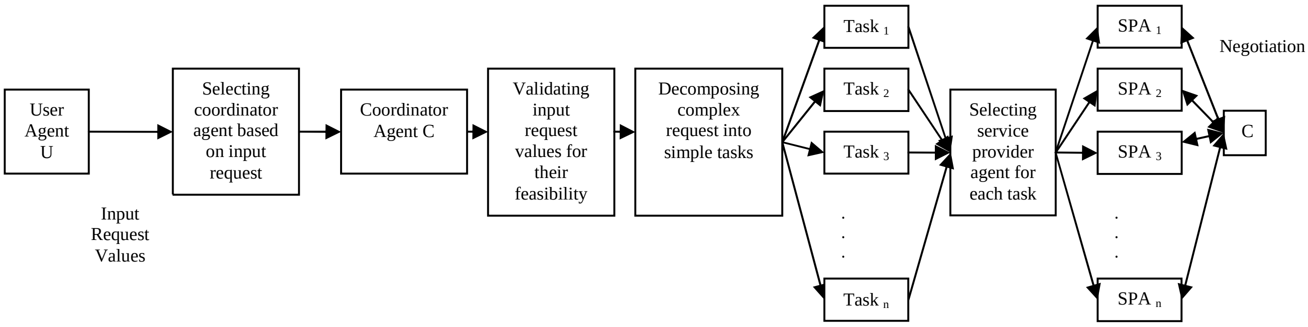

Figure 2(a): SWS Composition System with a dedicated coordinator agent

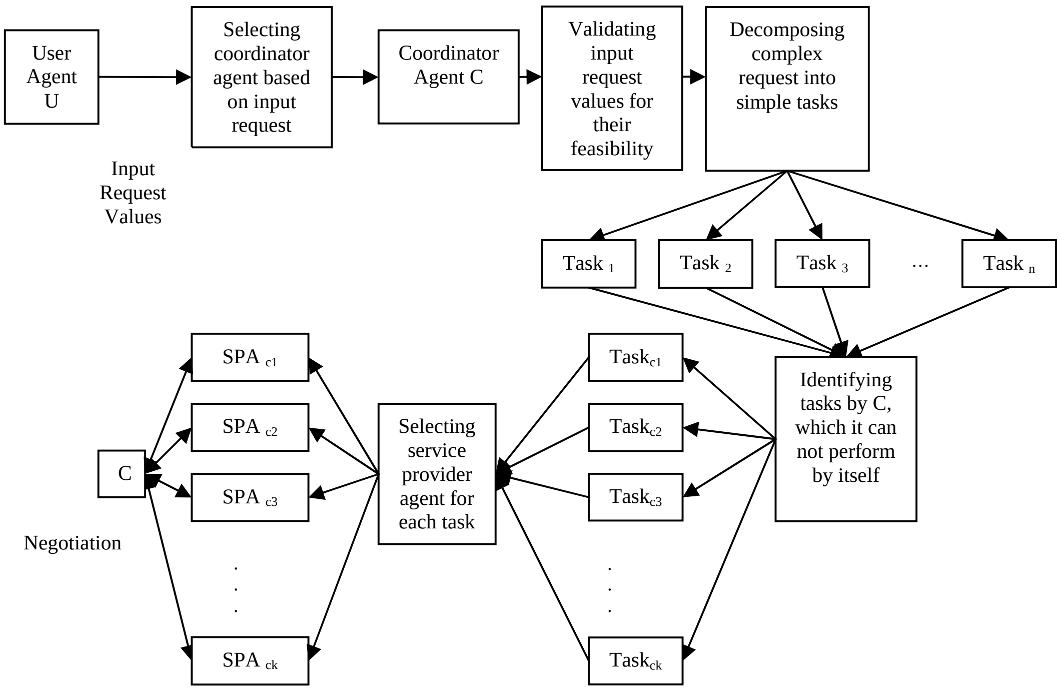

Figure 2(b): SWS Composition System with coordinator agent, which also perform some tasks

But in the second model, the coordinator agent can be made to check whether the constraints seems to be satisfiable up-to a level or are clearly unsatisfiable. So in case of unsatisfiablity, coordinator can warn the user agent U about it and there will be no need to move further in the composition process. However, in normal cases, this model will have more communication and computation time as compared to the first model, but it will have more reliability also.

3 A MAS Based SWS Composition Approach

In this section, we present a novel MAS Based SWS Composition Process (MABSCP). MABSCP is based on the model shown in Figure 2(a) and described in preceding section, which uses an independent dedicated coordinator agent for controlling the various activities in composition process. A layout of the MABSCP is shown in Figures 3(a) and 3(b). The system mainly consists of following three types of agents:

- Service Requester Agent (SRA)
- Coordinator Agent (CA)
- Service Provider Agent (SPA)

SRA has the responsibility to perform the request to CA. The request by SRA is then specified in the term of ontology, which is then used by the CA. An intelligent CA has following properties and capabilities:

- CA is a modular, self-contained software component wrapping coordination services, with ontological service description.
- It has the capability of validating the constraints, preferences, and other higher level parameters of the input request by the SRA.
- It has the capability of validating if the input activity is atomic or complex. In case it is complex, interpreting it as task comprising of various atomic activities of varying granularity and decomposing it into atomic tasks according to their ontology description.
- It can evaluate and assess the SPAs using their cognitive and Quality of Service (QoS) parameters.
- It can negotiate with the SPAs as well as SRA to adjust activity-input, SRA preferences and constraints, and to obtain matching common IOPE (Inputs, Outputs, Pre-conditions, and Effects), in order to satisfy the ultimate request.
- It makes arrangement for outsourcing the activity to SPAs based on FIPA Contract Net Protocol [12] and agent's communication interface built upon FIPA-ACL [4].

An intelligent SPA has following properties and capabilities:

- SPA is a self-contained, modular agent wrapping services in the form of software components, with the corresponding ontological service description.
- The purpose of SPA is decided by the services it wraps.
- It is able to understand the meaning of activity, it has to perform.
- SPA joins the composition process, only for the time its service is required.

Layouts of MABSCP are shown in Figures 3(a) and 3(b), respectively for CA and SPA selection. The system has novelty in the following aspects:

- In the practical scenario, it is not always the case that the negotiation can be performed freely between SRA and CA or SPA and CA (or SRA and SPA in reference of other model). Because there can be following conditions:
  - Some of the request parameters, constraints or preferences are so rigid that they are not negotiable.
  - All of the constraints, preferences and parameters specified by the customer are such that they are not negotiable.
  - Customer is not at all interested in performing the negotiation.

  Avoiding these conditions may cause un-necessarily wasting the negotiation effort. If the first condition is there, then the system can avoid these particular parameters for the negotiation process. For example, in some cases, the time constraints may be very rigid and not meeting it may cancel the request. To the best of our knowledge, no literature on MAS based SWS composition handling this issue is there. MABSCP has handled this problem, by communicating with the SRA regarding this, before starting the actual composition process.
- Various different requests from SRAs, even falling in same domain, can not be treated in the same manner. The performance of agent may vary depending on some parameters of requests. For example, in famous travel planning scenario, one agent may be performing better in handing the problems of travel planning in Europe region, while the other may be performing for Asia region. It may also be the case that an agent may be desired to perform for some parameters only for some particular time-period, while for other in other time-period. To the best of our knowledge, no literature on MAS based SWS composition handling this issue is there. MABSCP is handing this problem by dividing the requests falling in same domain into multiple categories and deciding the category in the related domain for input request of that domain. It further assesses only those agents for their

possible selection, who desires in performing for the decided category of input request, at that time. We have only applied this procedure for the selection of CA. However, it can be applied to the selection of SPAs in the similar fashion. But as the request to CA is in the form of a complex task, so there is more chances that it can be properly categorized, so this concept seems to fit in this case in better way.

- Request from SRA may contain such parameters, constraints or preferences which seem infeasible or within specified constraints the satisfaction of request seems near to impossible. In those cases, the composition will fail at the end, when even after negotiation between SPAs and CA, the constraints are not satisfied. This problem has already been discussed in detail in section-II. MABSCP handle this problem by placing validation on these parameters of requested task by the CA, before proceeding further in the process.

MABSCP involves sending a request from the SRA, user agent U, to the system, which is further represented in the term of ontologies.

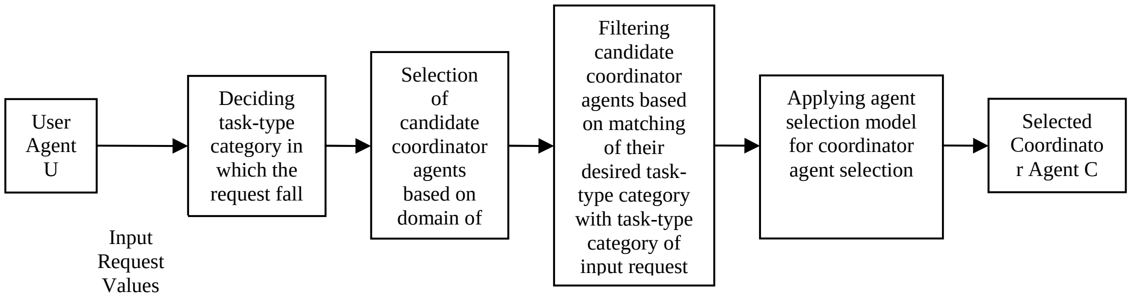

Figure 3(a): Coordinator agent selection

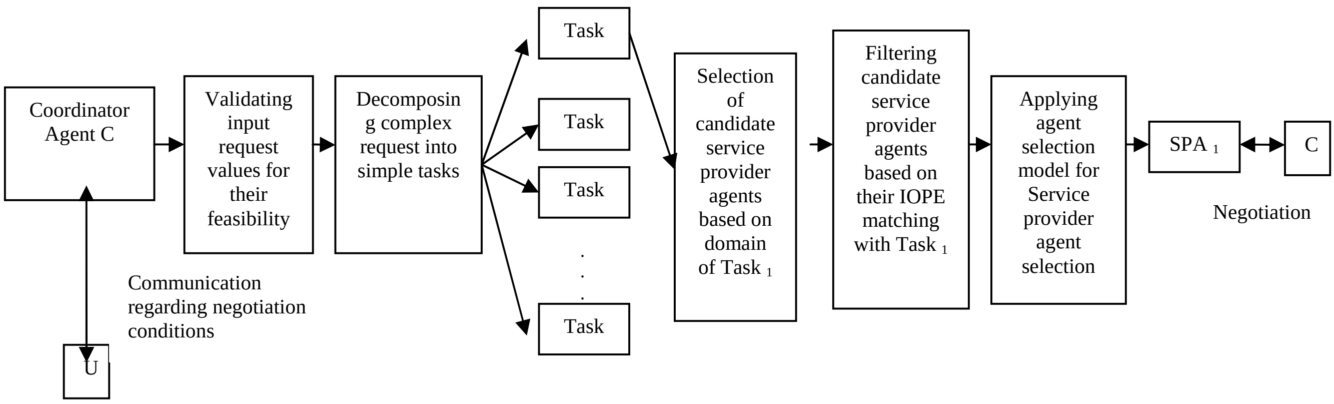

Figure 3(b): SPA agent selection and composition

The parameters in the request are used to decide the domain and further the task-type category within the domain of the request. The domain of request is used to discover the entire candidate CAs after matching from their published ontological service profiles. The candidate CAs are further filtered based on the matching that if the task-type category of the input request is matching with any of the category mentioned in the set of desired task-type categories of the candidate CA or not. The matchmaking here can be performed based on LARKS [15] or service discovery techniques based on UDDI protocol [8] or semantic matching [9] based on ontology profiles like in DAML-S [17], OWL [7] or using any other service matching method. Various techniques of discovery [14] can be applied here like keyword matching, controlled vocabulary matching, semantic matchmaking etc. The system can be made to proceed with the exact match, plug-in match, subsumption match, or intersection match as required in the process. An index of selection (IoS) is then calculated for each of the filtered CAs using an agent selection model and the CA with maximum IoS is selected as the coordinator C for given composition problem. At this stage, U can perform negotiation with C based on FIPA Contract Net Protocol [12]. C accept task from U by means of agent's communication interface built upon FIPA-ACL [4]. All of the remaining activities further in the process are coordinated by C; however, it can perform negotiation with U during the process, if required and allowed. C acquires the conditions of negotiation from U by means of agent's communication interface built upon FIPA-ACL [4]. Now, C performs a validation over the parameters, preference, and constraints to check their feasibility. It also determines from the input request, if it is an atomic activity or complex one. In case the input request is complex task, then C decomposes it into atomic tasks, $Task_1$, $Task_2$, $Task_3$ ... $Task_n$, of varying granularity. Now for each atomic task, the candidate SPAs are discovered and filtering over discovered SPAs are performed based on their IOPE matching with the required task. The matchmaking at this stage can be performed in the similar way as described above for the CA. IoS is then calculated for each of the filtered SPAs using an agent selection model and the agent with maximum IoS is selected as service provider. The negotiation between C and the selected SPA now can be initiated if required and allowed. Figure 3(b) shows this process for $Task_1$ only, however, it is performed for each atomic task in the same manner. The selection models here can be based on the various important attributes of SPAs like QoS (Quality of Service) attributes or Cognitive attributes. Various cognitive parameters are like capability, desire, intention, commitment, trust, reputation etc. and various QoS parameters are like cost, response time, reliability, accuracy, security feature, execution time, exception handling feature, penalty on breaking service contract etc.

4 Implementation Issues

This section discusses the implementation issues involved in the MABSCP, taking Education Planning Problem as the application. The education planning is a new problem for semantic web based system. No literature is there which explore this area for application of semantic web. The education planning problem for taking admission for higher education involves various activities like Counseling and Preparation, Institute Tracking, Admission Consultancy, Financing, Transportation Booking, and Map and Weather Information. All these activities can be assigned to different SPAs.

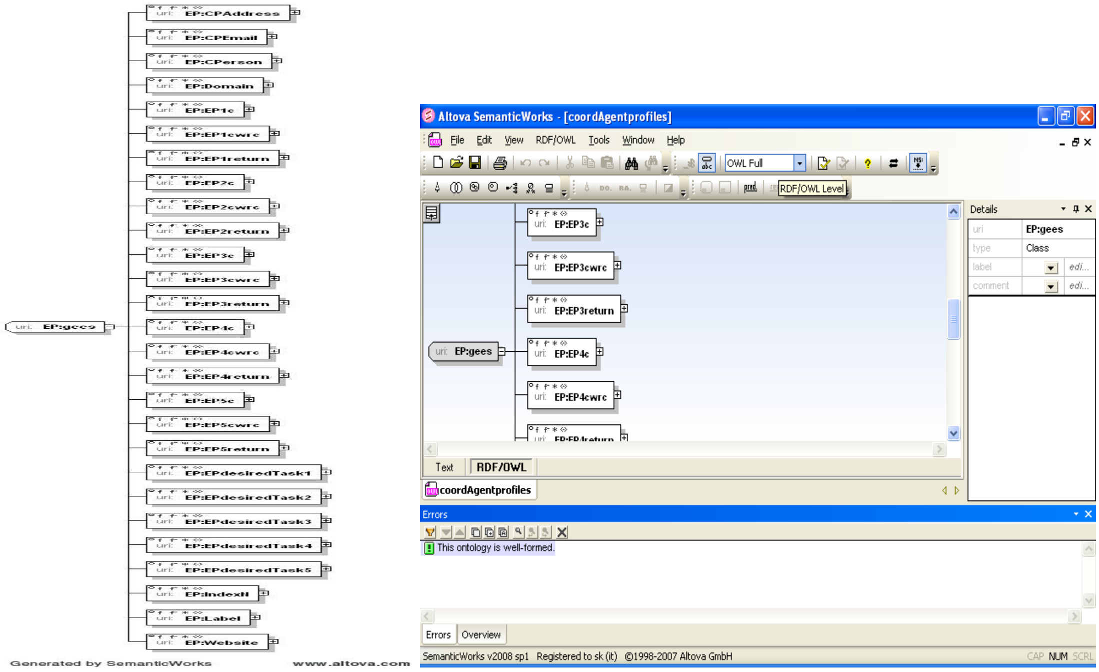

Figure 4: Coordinator Agent profile

The profiles of both coordinator as well as other task-specific agents can be developed using any semantic web service description language like RDF/RDF-S [5], DAML/DAML-S [2], OWL [7] etc. Some of the semantic web tools like Protégé [13], Jena [6] and Altova SemanticWorks [1] provide the support for developing the profiles in either RDF/RDF-S or OWL. Jena Ontology API provides an easy to use APIs (Application Programming Interfaces) for ontological profile development and good reasoning support. These profiles are then published on the web and can be accessed or manipulated by the semantic web service composition system. Structure of sample profiles prepared using Jena Ontology APIs and observed in Altova SemanticWorks for a coordinator agent for education planning is shown in Figure 4. Figure also shows that the ontology in the profiles is well-defined under OWL-Full RDF/OWL level. The reasoning in the system is performed using Jena's *OWLReasoner*. However, Jena also provides several *Reasoner* types to work with different types of ontology.

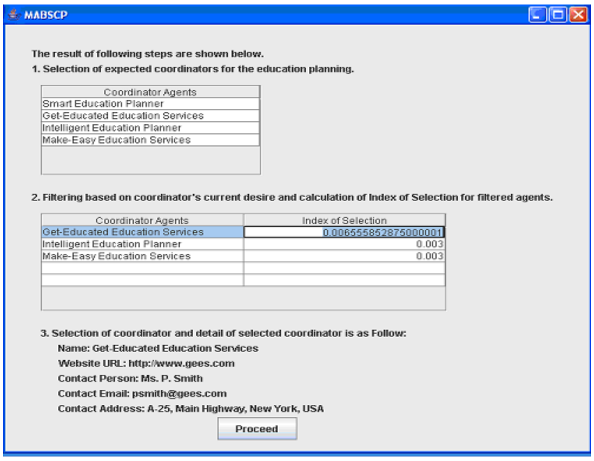

Figure 5: Coordinator Agent Selection

The system implemented using Java and related tools easily access the service profiles and uses the Jena APIs for interrogating, manipulating, or querying the profiles. The querying support provided by the Jena APIs, which is internally implemented in query language SPARQL [10] is used for querying over the profiles. The query language RDQL [11] can also be used with Jena for availing the advanced querying support. For handling the large service profiles the persistent ontologies of Jena ontology APIs can be used. The implemented system mainly uses the exact-match approach in discovery process. The composite input ontology in the system mainly have three components: Qualification Input like course in which admission is sought, entrance examination score, qualifying examination score; Additional Admission Requirements like session of admission, date of birth, gender; and Preference and Constraints like finance needed or not, map needed or not, budget constraint, travel class constraint etc. Then the further steps for the selection of a coordinator agent such as domain based filtering, agent's desire based filtering and rating using a cognition based mathematical indexing are implemented and a coordinator agent is selected. This agent then perform decomposition of the request based on input task ontology and a SPA for each task is selected using different steps from MABSCP like domain based filtering, IOPE filtering and selection based on their QoS and cognition parameters. The validation of input and communication with user-agent regarding negotiation conditions is also implemented in the system.

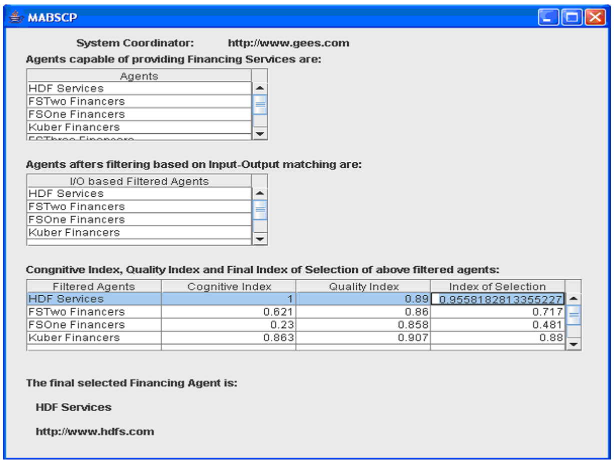

Figure 6: Finance Service Agent Selection

For elaborating the implementation, we have shown the output of different steps for selection of coordinator agent in Figure 5 and for selection of a Finance Service Provider Agent in Figure 6, otherwise these steps do not require user-interaction. As shown in Figure 5, the agent 'Get-Educated Education Services' has maximum index of selection and hence it is selected as coordinator agent. Figure 6 shows that agent 'HDF Services' with maximum index of selection get selected as Finance Provider Agent. Jade [16], a Java based agent-development environment, can be used to develop the agents and to establish communication between them. Jade also provide the environment for implementing the FIPA Contract Net Protocol [12] for negotiation between the agents involved in the system. Figure 7 shows the layout of composed system in the form of blocks, with each block representing different selected agents for various activities such as counselling and preparation service, institute tracker service, admission consulting service, transportation service, financing service, map & weather information service involved in the education planning request.

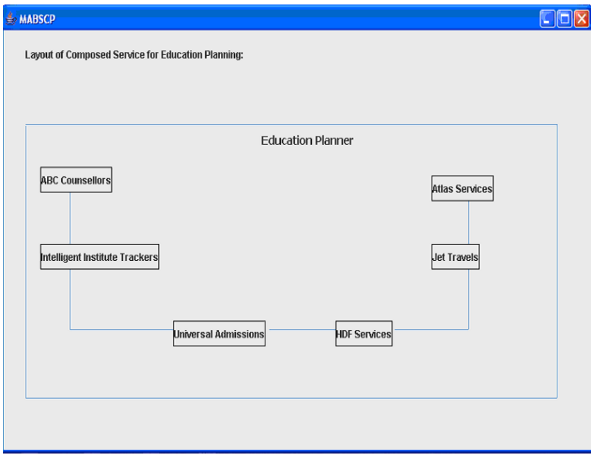

Figure 7: Layout of a composed system

These selected agents are now invoked to take the respective services and a FIPA-compliant agent communication interface is provided between them using Jade, so that they can communicate with each other.

5 Conclusion

The paper presented two models for multi-agent based semantic web service composition process, depending on the variations in the composition process. These models basically vary on the use of coordinator agents in the composition process. The model which uses coordinator agent found to be more advantageous than the other one. Further a SWS composition approach based on the presented composition model (with coordinator), MABSCP, is also presented. MABSCP deals with some of the important issues like handling negotiation conditions, validating input request, and categorizing input request based on its nature. The work also presents a new application area of semantic web technology i.e. education planning. Paper introduces this area with some related implementation by applying MABSCP on it. Our future work will involve exploring further the area of education planning as an application of semantic web technology and to further enhance MABSCP and also provide better selection approach for service provider agents.

References

[1] Altova. SemanticWorks Semantic Web tool - Visual RDF and OWL editor,http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html, Website accessed March 01, 2008.

[2] DARPA. The DARPA Agent Mark-up Language, http://www.daml.org/, Website accessed Feb 28, 2008.

[3] Ermolayev, V., Keberle, N., Kononenko, O., Plaksin, S., Terziyan, V. Towards a Framework for agent-enabled semantic web service composition, Int. Jl. Of Web Services Research, V. 1(3), p. 63-87, 2004.

[4] FIPA Architecture Board, Foundation for Intelligent Physical Agents. FIPA Communicative Act Library Specification, http://www.fipa.org/specs/fipa00037/SC00037J.html, Web-Site accessed on Feb 12, 2008.

[5] Herman, I., Swick, R., Brickley, D. Resource Description Framework (RDF), http://www.w3.org/RDF/, Website accessed Feb 28, 2008.

[6] HP Labs Semantic Web Programme. Jena- A Semantic Web Framework for Java, http://jena.sourceforge.net/, Website accessed March 01, 2008.

[7] McGuinness, D.L, Harmelen, F.V. OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features/, Website accessed on Feb 13, 2008.

[8] OASIS. OASIS UDDI Specifications TC - Committee Specifications, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm, website accessed Feb 13, 2008.

[9] Paolucci, M., Kawamura, T., Payne, T., Sycara, K. Semantic Matching of Web Service Capabilities,

In: Proc. Of Int. Semantic Web Conference (ISWC, 2002), Italy, 2002.

[10] Prudhommeaux, E., Seaborne, A. SPARQL Query Language for RDF, http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/ , 2008.

[11] Seaborne, A. RDQL - A Query Language for RDF, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/ , 2004.

[12] Smith, Davis. Foundation for Intelligent Physical Agents, FIPA Contract Net Interaction Protocol Specification, http://www.fipa.org/specs/fipa00029/SC00029H.html, Web-Site accessed on Feb 12, 2008.

[13] Stanford Center for Biomedical Informatics Research. Protégé, http://protege.stanford.edu/, Website accessed March 01, 2008.

[14] Stollberg, M., Haller, A. Semantic Web Services Tutorial, 3rd International Conference on Web Services (ICWS 2005), 2005.

[15] Sycara, K., Widoff, S., Klusch, M., Lu, M.J. LARKS: Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace, Autonomous Agents and Multi-Agent Systems, v. 5, p. 173-203, 2002.

[16] Telecom Italia Lab. Java Agent Development Framework, http://jade.tilab.com/, Website accessed March 01, 2008.

[17] The DAML Services Coalition. DAML-S: Semantic Mark-up for Web Services, http://www.daml.org/services/daml-s/2001/10/daml-s.pdf, Website accessed on Feb 13, 2008.