

## Multi-agent and Workflow-Based Web Service Management Model

Wenjia Niu, Quansheng Dou, Xu Han, Xinghua Yang, Zhongzhi Shi

► **To cite this version:**

Wenjia Niu, Quansheng Dou, Xu Han, Xinghua Yang, Zhongzhi Shi. Multi-agent and Workflow-Based Web Service Management Model. 6th IFIP TC 12 International Conference on Intelligent Information Processing (IIP), Oct 2010, Manchester, United Kingdom. pp.26-34, 10.1007/978-3-642-16327-2\_7. hal-01060355

**HAL Id: hal-01060355**

**<https://hal.inria.fr/hal-01060355>**

Submitted on 4 Sep 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Multi-agent and Workflow-based Web Service Management Model

Wenjia Niu<sup>1,2</sup>, Quansheng Dou<sup>3</sup>, Xu Han<sup>1,2</sup>, Xinghua Yang<sup>2</sup>, Zhongzhi Shi<sup>1</sup>

<sup>1</sup> Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, 100080, Beijing, China

<sup>2</sup> Graduate School of the Chinese Academy of Sciences, 100039, Beijing, China

<sup>3</sup> Shandong Institute Of Business and Technology , 264005, Yantai, China

{niuwenjia, douqsl, hanxu, yangxh,shizz}@ics.ict.ac.cn

**Abstract.** The coordination between agent service and Web service is the key factor for intelligent Web service management in the multi-agent based Web service framework. In view of the drawbacks of existing coordination approaches for agent service and Web service, this paper proposed a multi-agent and workflow-based Web service management model. Through analyzing the interaction relations between agent service and Web service in the logical action-based task environment, a uniform task view for intelligent Web service is built. And based on such task view, a workflow towards special task is designed to realize intelligent Web service discovery and cooperation and composition. This model provides a more flexible Web service management.

**Keywords:** Multi-agent; Workflow; Intelligent Web Service; Task View

## 1 Introduction

Web service technology has already been an interesting research problem in distributed computing field. Based on the Internet standard protocol, it aims to blend the heterogeneous applications, and realize interaction between different platforms. As the developing trend of the future Internet, its basic architecture is based on the interaction among the three roles: service provider, service broker and service requester. The service provider uses WSDL[1] to describe the service and publishes it to the service broker. The service requester submits request to the service broker, and uses SOAP[2] to invoke Web service. The service broker registers the services the provider released, and helps the requester search and assembly the services.

From the angle of role management, the traditional Web service has some shortages, e.g. lack of self-organization capabilities, possible resource conflicts during the execution and rigidified processing mode. Agent has intelligent characters such as autonomy, interaction and initiative etc., so it is very suitable to construct flexible and intelligent Web service management system with complex structures. Web service technology is complementary with agent technology, and the combination of such two technologies has now become a major research direction in the intelligent Web

service field. Although some work has been done on the combination of agent and Web service at home and abroad[3,4,5,6], but from the global aims of the intelligent Web service, those key issues about the semantics expression, service management and intelligent Web mining, have not been resolved very well. Especially in the intelligent Web service management field, how to realize flexible interactions between the Web service roles is currently a key problem which extraordinarily needs to be resolved.

In intelligent Web service management, many researchers have contributed helpful attempts and researches, and workflow as an important technology has also been tried in the coordination and composition of intelligent Web service, and typical work was described in [7,8,9,10,11]. Through the analysis of related work, there are two major methods for workflow to coordinate tasks. One is to directly coordinate Web service tasks with agent technology, while the other is to indirectly invoke Web service by coordinating agent tasks. These two methods proposed the idea of Web service management based on workflow from different angles. In fact, there exists tight inherent relevance between Web service task and agent task. Therefore, the two tasks are separated in processing, which will result in the lack of unified task views and not beneficial to the global workflow task coordination and flexible Web service management.

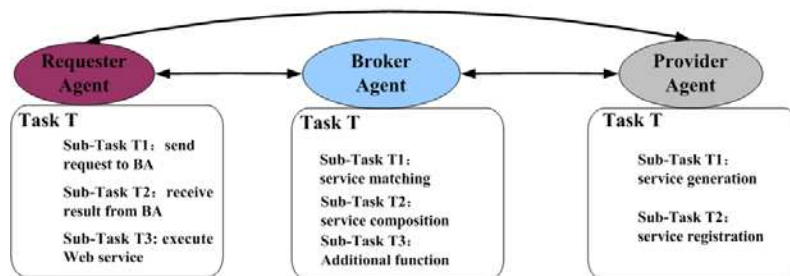
For the problem mentioned above, this paper built unified intelligent Web task view, and based on such view we designed the global task view oriented workflow to realize the intelligent management of Web service, which can be used to improve the global coordination and management capabilities of the Web service.

The remainder of this paper is organized as follows. Section 2 presents the intelligent Web task view. Section 3 presents the service workflow management. In Section 4, we give the intelligent Web service management model. Section 5 draws conclusions.

## **2 Intelligent Web Task View**

### **2.1 Agent Task**

There are three types of Agents in intelligent Web: Provider Agent (PA), Requester Agent (RA), Broker Agent (BA). PA supplies Web service, and RA requests Web service from PA, while BA helps RA to locate PA, find and combine corresponding services. The interactions between the three agents are shown in Fig. 1. From the angle of ultimate goal, these agents supply Web service for users by mutual negotiation. From the angle of role that the agent acts as, one agent can be either a service provider agent or a requester agent. From the angle of functional component, each agent encapsulates specific function. In the interaction process, each agent implements its own task for the global aim, and each task can be also divided into several sub-tasks, these tasks are called agent tasks in short.



**Fig.1: Multi-agent Interaction in the Intelligent Web**

Agent task environment can be formalized as a two-tuples  $\langle P; R \rangle$ .  $P = \{A, T, \text{Sub-T}\}$ , where  $A$  represents agent,  $T$  represents abstract tasks of agent, and  $\text{Sub-T}$  represents sub-tasks after the division and instantiation of abstract task;  $R = \{E, F\}$ , where  $E = \langle A, T \rangle$  represents the abstract task  $T$  that agent  $A$  needs to realize;  $F = \langle T, \text{Sub-T} \rangle$  represents that abstract task  $T$  can be implemented by an instantiated single  $\text{Sub-T}$  or composite  $\text{Sub-Ts}$ . The above formalized description shows that it is easy to translate the invoke and operation of agent task into the function realization in the program. For example,  $RA.T.Sub-T2$  represents that  $RA$  invokes its  $\text{Sub-T } T2$  to obtain the results returned by  $BA$ .

## 2.2 Web Service Task

Web Service can be defined as a group of service access points through WSDL description, and requester can access the service through these points. The WSDL service file, first abstractly describes the access operation, the request and response messages, which will be bound into concrete transfer protocol and message format. WSDL includes service interface and service realization. Analysis from the functional structure angle, the execution of each concrete service means the implementation of one task. For example, the task of ticket booking service is to realize the users' ticket booking request, and this task can be divided into several sub-tasks, such as ticket query task and cash settlement task etc.. In addition, for the software development, Axis2[12] has already realized the bidirectional auto generation both from Java class to WSDL and from WSDL to Java class, which means that Web service is a functional entity which can execute the task just like a Java program.

Except the WSDL, the Semantic Web Service technology uses the semantics description language (e.g. OWL-S[13]) which can be understood by the computer, and enrich the service semantics with domain ontology. It aims to intelligently operate the service through inference function of the ontology. OWL-S describes a Web service by describing Service Profile, Process Model and Service Grounding, in which Service Profile describes the input, output, precondition and effect (IOPE). As

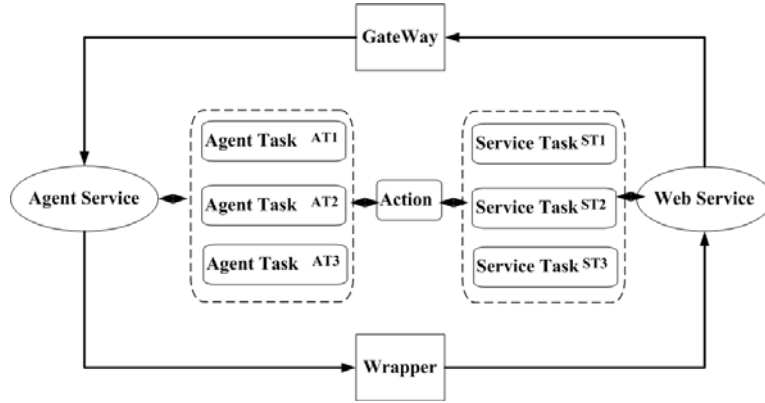
description of the service function attributes, in fact IOPE also describes the task which the Web service functional entity will execute.

The above analysis shows that Web service task just means the function execution of the Web service. Take the suggested description language OWL-S of Web service as standard, with IOPE to describe the service function, the task environment of Web service can also be formalized into a two-tuples  $\langle P; R \rangle$ .  $P = \{S, T\}$ , where S represents Web service, T represents Web service task;  $R = \langle S, T \rangle$  represent the task T that Web service S will implement; and  $T = \{I, O, P, E\}$ , which represents that if the input and precondition are satisfied, executing task T can obtain the output and effect of the service. That is the task execution will change the current status  $\{I, P\}$  into the effect status  $\{O, E\}$ .

### 2.3 Task View

From the point of view of theory and engineering implementation, the agent and Web service have already overcome the description and communication barrier in the integration aspect in JADE, which provides a basic infrastructure for the relation mapping between the agent task and Web service task. Nguyen proposed WS2JADE framework[14], which realized the encapsulation from Web service to agent service through GateWay technology. Varga discussed the compatible problem between the FIPA ACL on JADE platform and the Web service protocol, and utilized Wrapper mechanism to encapsulate the agent service of the JADE platform into the standard published Web service. Furthermore, it makes the Web service invoke agent service and return corresponding results.

Agent service description language SDLSIN[16] is the improved edition based on CDL, SDL and LARKS. It clearly defines that agent service is composed of several tasks, i.e.  $AS = \bigcup_i (\text{Action:Concept})_i (i \geq 1)$ , where Action is the action expression in dynamic description logic (DDL)[17]; Concept is the concept expression. Agent service executes the corresponding agent task through invoking the logic action. According to WSDL, an atomic Web service is composed of one or multi operations (tasks), i.e.  $WS = \bigcup_i WS\_Ti (i \geq 1)$ , Shi[17] has proved that Semantic Web Service is equivalent to logic action in semantics. The execution of corresponding Web service task is equivalent to the execution of action in DDL. Through above analysis, the interaction of agent service and Web service, and the action logic of the task together set up a bridge for the unified logic view of Web service task and agent task(See Fig.2).



**Fig.2: Logic Relations between Agent Task And Web Service Task**

In fact, agent service and Web service formed the unified task view based on action. This task environment can be formalized as a two-tuples  $\langle P: R \rangle$ .  $P = \{AT, ST, Action\}$ , where AT represents agent task, ST represents Web service task, action represents logic action of the task;  $R = \{E, F, G\}$ , where  $E = \langle AT, Action \rangle$  represents logic action the AT will implement;  $F = \langle ST, Action \rangle$  represents logic action the Web service will implement; G represents the relation between logic action  $\pi$  of agent task and logic action  $\pi'$  of Web service (See Table 1).

**Table1: Relations among Logical Action of Task**

	Logic Description	Explanation
Sequence( $\pi, \pi'$ )	$\pi ; \pi'$	$\pi$ and $\pi'$ execute sequentially, i.e. the execution effect of $\pi$ is the execution precondition of $\pi'$
Choice( $\pi, \pi'$ )	$\pi \cup \pi'$	Either $\pi$ or $\pi'$ is chosen to execute, i.e. $\pi$ is similar with $\pi'$ , choose one to execute
Any-Order ( $\pi, \pi'$ )	$(\pi ; \pi') \cup (\pi' ; \pi)$	$\pi$ and $\pi'$ execute sequentially, $\pi'$ and $\pi$ execute sequentially, one of the compound action $(\pi ; \pi')$ and $(\pi' ; \pi)$ is chosen to execute.

### 3 Service Flow Management

Workflow is a managing process which organizes a group of tasks to implements some special function. Each task is implemented by one or more programs. The trigger sequence and condition are defined in the workflow, which implements the task trigger and synchronization as well as the workflow transmission. Intelligent Web service workflow management is to realize the workflow definition and

management for the task of agent service and Web service, and propel the execution of the task according to the logic defined in advance. The whole workflow management model needs to consider the following issues. First, control role design to control and supervise the execution of workflow. Second, workflow design is to realize the Web service discovery and composition, and automatic or semi-automatic invoke and design and organize the subtasks. Third, conflict resolution is to resolve the conflicts of synchronous and asynchronous cooperation among the tasks.

### 3.1 Design of Control Role

The control role needs to control all the resources. In intelligent Web service, BA has features to easily interact with other agent to get the global view of the resources. BA with workflow control structure aims at implementing the message transferring between the workflows as well as the task execution in the workflow, so as to intelligently control the workflow. The major design has the following five modules: communication interface module, workflow supervision module, policy repository, chief control module and status repository. Communication interface module is responsible for message transmission and communication between the workflow tasks. Currently for the message transmission mechanism on the MAGE platform, in order to avoid marshalling and unmarshalling procedure, message is coded into Java object rather than string to transmit. When passing the platform border, message is automatically transformed to FIPA compatible grammars, codes and transfer protocols. Policy repository mainly stores the policy of executing the corresponding task in the workflow according to the current status, e.g. conflict resolution policy etc.. The workflow status supervision module mainly supervises the execution status of the task and conflict generation. The chief control module works like the nerve center to coordinate the execution of other modules. The status repository is constructed as a form, and mainly stores the execution status of the tasks.

### 3.2 Workflow Design

The workflow of Web service mainly includes service discovery workflow and service composition workflow. Service discovery is the precondition and foundation of service composition, so in the whole workflow, service composition generally includes discovery workflow. As shown in the right part of Fig.3 : ①agent task AT0 receives request from user, and decomposes request into several sub-goal, then distributes them to other agent task; ②Agent task invoke Web service task to realize the sub-goal based on the sequence relation between tasks; ③if agent task doesn't find suitable Web service or execution exception, then it will start other agent task and reduce the service discovery standard, then continue discovering task until find similar Web service task based on the Choice relation between tasks; ④Agent task combines the execution of Web service task; ⑤return execution results to users; ⑥if the composed Web service task executes with exceptions or errors, then ask AT0 to decompose the goal, and repeat the flow. The whole workflow shows that what the ②③ done is in fact to realize the service discovery.

The logic decomposition of tasks in step① can be done automatically by plan inference based on DDL actions. The plan inference can be simply defined as the following: if and only if action sequence satisfies goal  $\varphi$  for the ABox, RBox and ActionBox, then the action sequence is a plan to achieve the goal, the concrete inference algorithm is described in document[17]. As shown in the left part of Fig. 3, DDL inference machine generates the action plan through atomic inference. By atomic mapping algorithm and manual variable assigning mapping, finally realize atomic task workflow deployment.

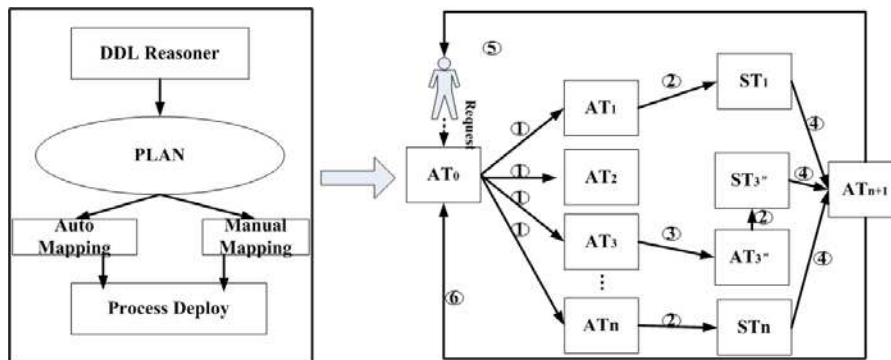


Fig.3: Workflow Design for Web Service Composition

### 3.3 Conflict Resolution

For the asynchronous cooperation of the task, the conflict generally occurs in the circumstance that the resources are not released in time after task execution. We mainly discuss the conflict resolution methods for synchronous cooperation of the tasks, and coordinate these synchronous tasks so that they can get the required resources, and insure that the tasks can execute fluently. Based on above design of control roles and workflow, the simple and effective coordinate algorithm is described in the following:

(1)Workflow supervision module supervises the task execution through the communication interface. When detecting resource operating conflicts during the execution process of two tasks, supervision module will inform the chief control module to instantiate a coordinating task (Co-Task) and an inference task (R-Task).

(2)The Co-Task will collect the status and resource information of the conflicting tasks, and submit to the global status repository as a form.

(3)Combined with the information of policy repository and global status repository, the inference task will infer and compute, and return the result to the chief control module.

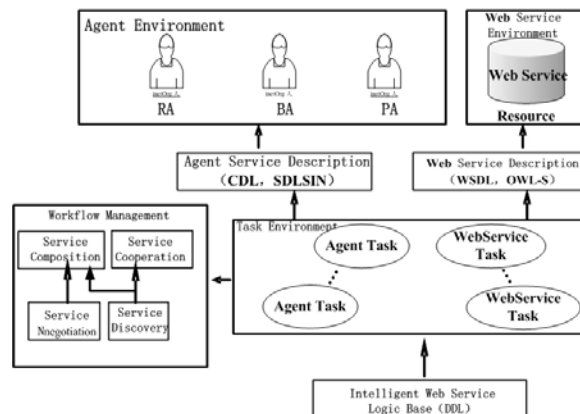
(4)According to the returned result, the chief module will restart one conflicting task and set waiting time for the other conflicting task through the communication interface.



## 4 Intelligent Web Service Management

The task-oriented intelligent Web service management model is to map the agent and Web service logic into the unified task environment, through the operation of workflow on the task, realize the service discovery and composition etc.. As shown in Fig. 4, the bottommost layer is the logic foundation of intelligent Web service, which mainly includes dynamic description logic and action theory etc., and the main goal of importing action is to represent and infer the static and dynamic knowledge in the intelligent Web according to the basic variation features of the dynamic world, and then provides the logic foundation for the unified view construction of task environment in the upper layer. The intelligent Web service description combines agent service and Web service description. Map operations of the two services into operations of tasks, so as to be invoked and managed directly by workflow. Workflow management module is composed of four parts: discovery module, negotiation module, cooperation module and composition module. Discovery module mainly judges if the providing service can satisfy the requesting service. Negotiation module mainly implements the interaction between the agents to achieve the consistent protocol by the operations on the tasks. Composition and cooperation modules are based on the service discovery, what they need to resolve is to satisfy the requester's request by the "reasonable organization" of Agent task and Web service task.

The unified-task-view-oriented workflow management module is the central part of the module. On the hand, the task abstract of agent and Web service forms the unified task view which provides a new visual angle and flexible process mechanism for the Web service management. The workflow designers don't need to consider the complex questions such as the role variation of the agent, the isomerism of Web service as well as the interaction of agent service and Web service etc., but emphasize on how to coordinate the execution of the tasks to achieve the goal in the environment with several tasks. On the other hand, the workflow and task are two entities closely related to each other. Unified task view enriches the task type and quantity that the workflow can coordinate to a certain extent with the automatical workflow design based on inference. The workflow design is now more flexible, which makes the Web service management module press close to industrial application to a higher degree.



**Fig.4: Intelligent Web Service Management Model**

## 5 Conclusion

This paper proposed a new Web service management model, by abstracting the unified task view of agent service and Web service, specific-task-oriented workflow management method was studied. But this model has many aspects which need to be improved further, e.g. further study the coordination method and complex conflict resolution method of the task, and further refine the operating steps of the workflow management etc.. In addition, refine the mapping rules of the task, and how to tightly combine with the industrial standard BPEL is the issue which needs to be studied further in the future.

**Acknowledgments.** This work was partially supported by the National Natural Science Foundation of China (No. 60775035, 60970088, 60903141), the National High-Tech Research and Development Plan of China (No. 2007AA01Z132), National Basic Research Priorities Programme(No. 2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06), Dean Foundation of Graduate University of Chinese Academy of Sciences(O85101JM03).

## References

- [1] Christensen, E., Curbera, F., Meredith, G. et al.. Web services description language (WSDL) 1.1, W3C note. Tech. Rep., W3C, 2001.
- [2] Box, D., Ehnebuske, D., Kajuata, G., Layman et al.. Simple object access protocol (SOAP) 1.1. Tech. Rep., W3C, 2000.
- [3] Gibbins, N., Harris, S., Shadbolt, N.. Web Semantics: Science, Services and Agents on the World Wide Web. WWW'03, 2003, 141-154.
- [4] Kumar, S., Mishra, R.B.. Multi-Agent Based Semantic Web Service Composition Models. INFOCOMP Journal of Computer Science, 2008, 7(3):42-51.
- [5] Payne, T. R.. Web Services from an Agent Perspective. IEEE Intelligent Systems, 2008, 23(2):12-14.
- [6] Jia, Y., Sun, Y., Shi, Z.. Research on Service Model of Multi-agent System. Computer Engineering, 2006, 32(22): 89-90.
- [7] Blanchet, W., Stroulia, E., Elio, R.. Supporting Adaptive Web-Service Orchestration with an Agent Conversation Framework. ICWS'05, 2005, 11-15.
- [8] Arpaci, A. and Bener, A.. Agent Based Dynamic Execution of BPEL Documents. ISICIS'05, 2005, LNCS, V(3733):332-341.
- [9] Ricci, A., Omicini, A., Denti, E.. Virtual enterprises and workflow management as agent coordination issues. International Journal of Cooperative Information Systems, 2002, 11 (3-4), 355-379.
- [10] Buhler, P. A., et al., Towards adaptive workflow enactment using multiagent systems, Information Technology and Management Journal, 2005, 6(1), 61-87.

- [11] Dong, W. Multi-agent test environment for BPEL-based web service composition. ICSOC'08 , 2008, 855-860.
- [12] <http://ws.apache.org/axis2/index.html>
- [13] The OWL Services Coalition. OWL-S: Semantic Markup for Web Services, <http://www.w3.org/Submission/OWL-S/>, 2004.
- [14] Nguyen, T., et al.. WS2JADE: Integrating Web Service with Jade Agents, SUTICT-TR, 2005.
- [15] Varga,L.Z., Hajnal,A.. Engineering Web Service Invocations from Agent Systems, CEEMAS'03, 2003, LNAI V(2691):626-635.
- [16] Jiang,Y., Shi, Z., Zhang, H., et al., Dynamic Service Matchmaking in Intelligent Web, Journal of Web Engineering , 2004,2(3):131-147.
- [17] Shi, Z., Chang, L.. Reasoning About Semantic Web Services with an Approach Based on Dynamic Description Logics. Chinese Journal of Computers, 2008,31(9):1599-1611.