International Journal of Computer and Communication Technology

Volume 1 | Issue 4 Article 12

October 2010

A Goal Driven Framework for Service Discovery in Service-Oriented Architecture: A Multiagent Based Approach

Ujjal Marjit
CIRM, University of Kalyani, West Benagal, India, sic@klyuniv.ac.in

Arup Sarkar
Departmentt of CSE, University of Kalyani, West Benagal, India, tellarup@gmail.com

Subhrangsu Santra
Department of RDM, University of Kalyani, West Benagal, India, ssantra@klyuniv.ac.in

Utpal Biswas
Department of CSE, University of Kalyani, West Benagal, India, utpal01in@yahoo.com

Follow this and additional works at: https://www.interscience.in/ijcct

Recommended Citation

Marjit, Ujjal; Sarkar, Arup; Santra, Subhrangsu; and Biswas, Utpal (2010) "A Goal Driven Framework for Service Discovery in Service-Oriented Architecture: A Multiagent Based Approach," *International Journal of Computer and Communication Technology*: Vol. 1 : Iss. 4 , Article 12.
DOI: 10.47893/IJCCT.2010.1059
Available at: https://www.interscience.in/ijcct/vol1/iss4/12

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

A Goal Driven Framework for Service Discovery in Service-Oriented Architecture: A Multiagent Based Approach

Ujjal Marjit
CIRM
University of Kalyani
West Benagal, India
sic@klyuniv.ac.in

Arup Sarkar
Departmentt of CSE
University of Kalyani
West Benagal, India
tellarup@gmail.com

Subhrangsu Santra
Department of RDM
University of Kalyani
West Benagal, India
ssantra@klyuniv.ac.in

Utpal Biswas
Department of CSE
University of Kalyani
West Benagal, India
utpal01in@yahoo.com

Abstract-**Automated service discovery is one of the very important features in any Semantic Web Service (SWS) based framework. Achieving this functionality in e-resource sharing system is not an easy task due to its hugeness and heterogeneity among the available resources. Any efficient automated service discovery will remain worthless until discovered services fulfill the required goal(s) demanded by the user or the client program. In this paper we have proposed a goal driven approach towards an automated service discovery using Agent Swarm in an innovative way .A novel multi agent based architecture has been introduced here for service discovery. Communications among the agent in service-oriented framework for the said purpose has also been illustrated here. Finally, the pictorial view of the running agent in the system is shown.**

Keywords-Web Services, Semantic Web Services, Agent, Service-Oriented Architecture

I. INTRODUCTION

Today the size of Internet and the web is increasing with a spell-bound speed. Use of them seems to be enormous and boundless. With the increasing size and utilization, they also inherited heterogeneity. Now a day's Internet and web are not only mean to provide pool of information or applications to the end users but also to provide e-services with the help of Web Service technologies. So naturally a huge trend about this technology can be seen from both the world of business and academy. Lots of investment already initiated in this domain. Actually web service technology brings the service oriented nature to the previously static information and document oriented internet based business world. Web Service technology was a revolutionary step towards the next generation of web and web related initiatives like – e-business, e-commerce, e-governance, e-learning etc. with a highly appreciated assurance of advance service oriented architecture based automated system; a system consists of machine interpretable information, service, and based on open standard based platform independent technologies.

It is known that every problem has a solution (in most of the cases), but it cannot depend on a single solution, no matter how advanced it is, if our world is changing every moment. Same sort of problem can be seen in web service based technology's also. The problem is, web services works well within a limited range of heterogeneity and provides only limited rigid set of services. So in response to the current state of demand for an automated system within an everyday changing heterogeneous environment, we moved to the Semantic approach of the services, i.e. Semantic Web Services (SWS).

An efficient service discovery mechanism is always a highly recommended and expected feature in any SWS based platform. But any service discovery mechanism is called efficient when they are totally able enough to find out exactly those services based on the user's desires which are faithful to achieve the requested goal successfully. Lots of research work is already done in this field to fulfill these needs. Many different Semantic Web Service (SWS) based framework has been proposed like, OWL-S, METEOR-S, WSMO etc., but every framework has their own kind of advantages and disadvantages. A recent interest of the research commodity can be found in the field of Multiagent System (MAS) based services. Though the agent related thoughts already attracted the vision of the researchers a decade ago, but the MAS based Web

Service discovery, selection, integration is a very adolescent initiative towards the future of web and web related technologies.

Here in this paper we have tried to introduce a novel framework for the automated service discovery based on the given goal. The most encouraging point of this work is based on MAS technology. A swarm of agents is fully responsible behind the scene from taking input from the end user or programs or machines to the final Web service selection as per their need.

The rest of the paper is organized as follows: Section-2 gives a brief discussion on the related research work. Section-3 illustrates motivation behind the work. Section-4 depicted the proposed architecture. Section-5 describes about the proposed framework and finally in Section-6 the conclusion and a note on the future work are stated.

II. RELATED RESEARCH WORKS

Lots of initiative already taken by many researchers in the agent-oriented field. Agent Oriented Programming shows one of the latest path, that developers are choosing to take the advantage of fully dynamic, robust, platform-independent and loosely coupled system. Actually agents are intelligent enough to take the decision according to the changing environment and changing level of available information which can not be expected from the traditional system.

A. G. Neiat et al proposes a new framework for an agent based Semantic Web service discovery [15]. A brokering system is maintained to avail the necessary interoperability between the semantic web service provider and the agents. They used the translation mechanism for the both the WSDL description of an Web service and the DF description of FIPA multi agents. Whereas the [16] propose a web service discovery depending on the level of cooperation between the UDDI registry and the DF the very own registry system of the FIPA standard based Multiagent system described in [16].

The authors of [17] break their work into two parts. At first they have proposed a technique to translate the contents of different ontology and content languages like WSML to OWL, WSML to FIPA SL, FIPA SL to WSML or OWL etc. in the second part they have proposed an architecture for semantic interoperability. The central part of this architecture is an Ontology Agent which is basically responsible for the all kind of communication between FIPA agents, and web services based on OWL and WSML.

[18] represents a Gateway architecture which allows Web Services to invoke the agent services and vice versa. It enables transparent, automatic connectivity between Web Services and agent services by preventing the risk of an asymptotic, divergent relationship developing between these two complementary technologies. At present the fully implemented version of this work called *Web Service Integration Gateway Service (WSIG)[19][20]* is available for download from the add-on page of the official JADE site.

III. MOTIVATION

Web Service is a technology meant to avail any service over the net by using platform and vendor independent open standards [1][2][3]. Most advantageous side of the web service technology is it can be programmed by almost any programming language, and can be run on any operating systems. They can easily be communicated by sending messages using universally accepted standardized XML messaging systems. Their modular design, language transparency and self describing nature encouraged lots of people to work on it. A typical web service is generally based on three key technologies WSDL[4], SOAP[5] and UDDI[6]. But we know they also facing some limitations as stated in section-1.

On the other hand Semantic Web ensures that the information stored into the web will be meaningful to the machines. The key reason behind this is the use of Ontologies which holds the semantic information of the sites, web pages etc.

The combination of Semantic Web with Web Service technology brings the same functionality to the web services as like the semantic web. Thus this new hybrid technology called Semantic Web Service promises more intelligent, automated, dynamic, and loosely coupled, machine accessible and understandable services without any hesitation or problems. But in reality, the semantic web service based systems are still lacks of standard and reliable service discovery, selection and invocation mechanism as we demand in theory. Machines or programs are still not efficient enough to take the automatic and dynamic decision for semantic web service based service discovery, selection and invocation. To wipe out this gap between theoretical

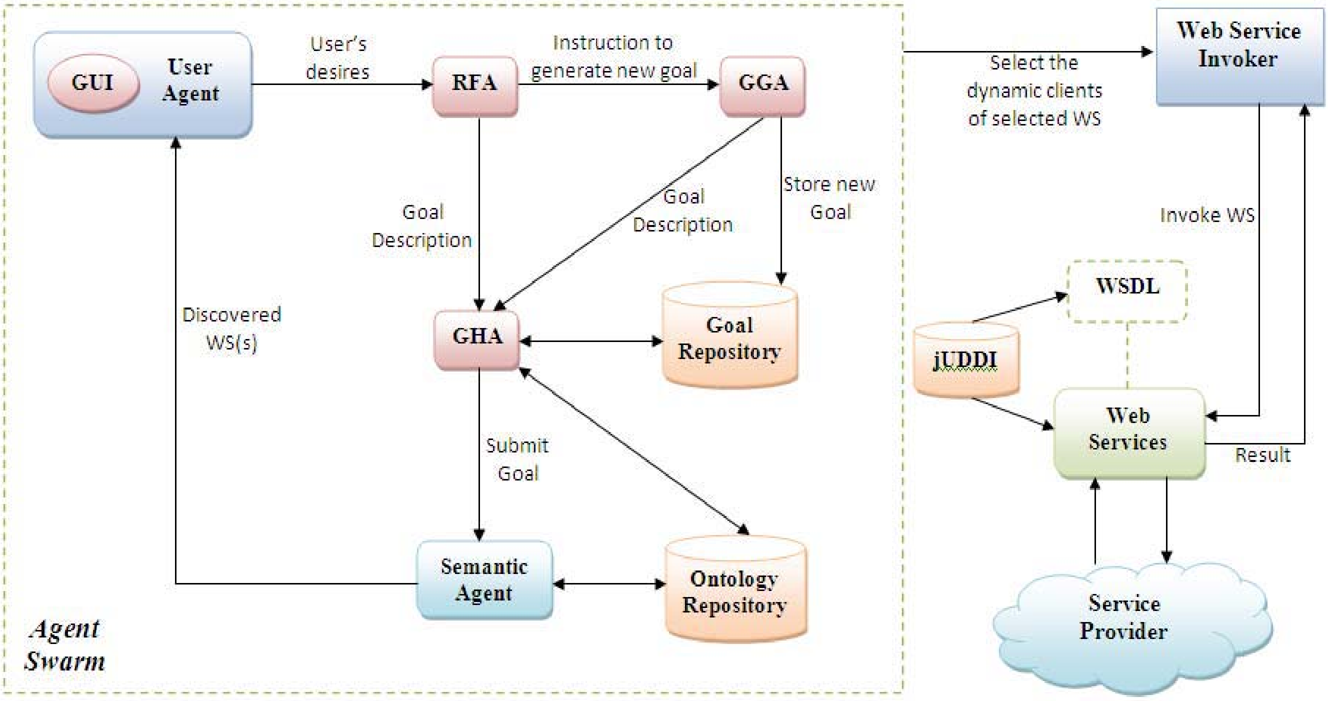


Fig.1 Proposed Architecture

claim and practical experience, Multiagent based system (MAS) is proven as a very useful tool or technology.

The Foundation for Intelligent Physical Agents (FIPA) is basically liable to maintain the standards of specification for MAS. It is an IEEE standards committee. All the promotion of the specifications for MAS related technology is generally governed by this non-profit organization.

IV. PROPOSED MULTIAGENT BASED GOAL DRIVEN FRAMEWORK FOR SERVICE DISCOVERY

We already illustrated the motivation behind this work in section-3. Architecture of the proposed system is depicted in the fig.1. Here in this section we are going to give a more detailed description of our architecture, a Multiagent based goal driven framework for service discovery. The main part of this architecture is its *Agent Swarm*. Agent Swarm comprised of five following different software agent namely User Agent, Request Forwarding/Filtering Agent (RFA), Goal Generator Agent (GGA), Goal Handler Agent (GHA) and Semantic Agent (SA). Each agent is introduced here with their own distinct roles. Each of them is described below:

A. **User Agent (UA):** This is the simplest natured agent throughout the framework. The main purpose is to collect information from client programs/end users/the machines using a GUI, web services or some external or internal agents. This agent also collects the list of finally discovered web services from the Semantic Agent (SA) and transfer and display the result to the user.

B. **Request Filtering/Forwarding Agent (RFA):** This agent is responsible for accepting the entire request message from the User Agent (UA). It then checks the validity of the message. Another checking is done just next to the validation checking against each of the valid message. This checks, is there any goal already exists in the goal repository that matches the user requirement i.e., the input collected from User Agent. If a match found then it collect all the necessary information about the goal file and then handover these information along with the user submitted query or request to the Goal Handler Agent (GHA). But if no match found then an instruction along with the user's request is submitted to the Goal Generator Agent (GGA) to generate a fresh new goal.

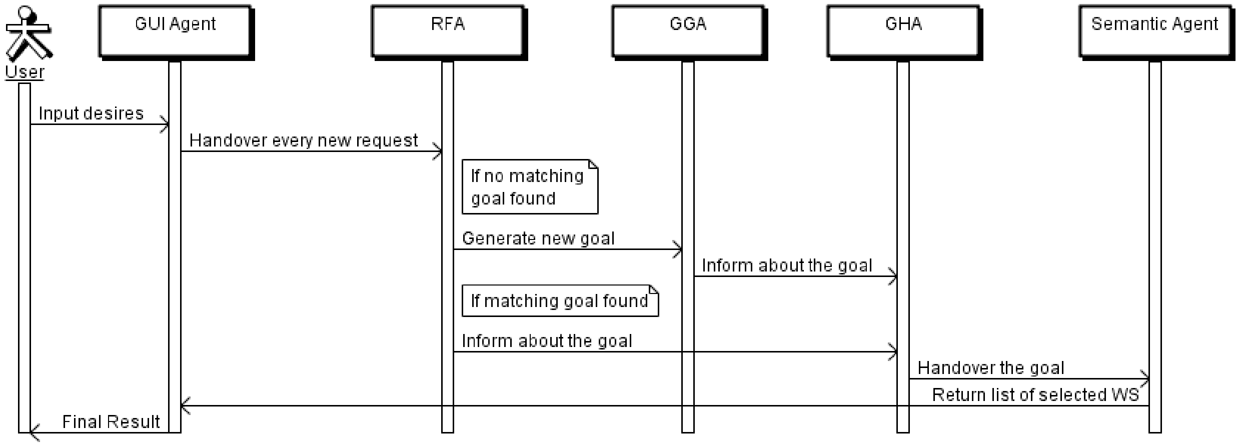
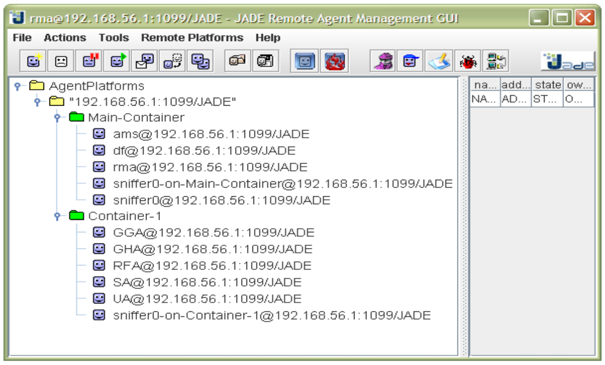


Fig.2 Sequence diagram of the Multiagent Communication in proposed architecture

C. **Goal Generator Agent (GGA):** This agent is accountable to generate a completely new goal based on the user's desires as transferred by the RFA. After a successful generation of a new goal, it will store it into the internal goal repository for further and future uses. In the next step it will hand over all the necessary information about the goal combined with user submitted information for further processing.

D. **Goal Handler Agent (GHA):** The selected goal by RFA or the generated goal by the GGA is finally processed by this agent. This agent's responsibilities are the important one comparing to the RFA and GHA. It basically selects the goal file to submit it to the Semantic Agent for matchmaking. But before it, two most important jobs it performs are: i) It checks the compositeness of the goal file. In simple words it checks either it is a composite goal or not. If it is a composite goal then try to break it into more than one simple goal file for an efficient semantic matchmaking and also keep the record of this process. ii) It performs a preliminary syntax based matching with the web service description. This step is important and very useful since it narrow the range of the web services for matchmaking, and also reduces the possibility of unnecessary matchmaking, for example, if we are looking for a medical service, then it ensures that we will not waste our time to try to make a match with the description of a car selling service.

E. **Semantic Agent (SA):** This is another very important agent and the last one among the Agent Swarm. The main part of the job of matchmaking is done here. It is the Semantic Agent's responsibility to successfully perform the discovery of web services based on semantic matching using available ontologies to it. It also maintains a record of the previous and current job of matchmaking. Suppose a new goal is submitted. The Semantic Agent will check the job records first for any previously matching against the submitted goal(s). If no record found a fresh matching is to be done. Suppose there is a previous record found, then it will check the any new semantic description of web service(s) is submitted into the Ontology Repository or not. If new ontology(s) found a new matching will be done and the record will be updated, otherwise the old list of the matched web service will be selected from the job record and return back to the User Agent as final result.


rma@192.168.56.1:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help

```
AgentPlatforms
  "192.168.56.1:1099/JADE"
    Main-Container
      ams@192.168.56.1:1099/JADE
      df@192.168.56.1:1099/JADE
      rma@192.168.56.1:1099/JADE
      sniffer0-on-Main-Container@192.168.56.1:1099/JADE
      sniffer0@192.168.56.1:1099/JADE
    Container-1
      GGA@192.168.56.1:1099/JADE
      GHA@192.168.56.1:1099/JADE
      RFA@192.168.56.1:1099/JADE
      SA@192.168.56.1:1099/JADE
      UA@192.168.56.1:1099/JADE
      sniffer0-on-Container-1@192.168.56.1:1099/JADE
```

na...	add...	state	ow...
NA...	AD...	ST...	O...

Fig.3: Agents running

There is a component in our proposed architecture called ***Web Service Invoker*** which can be used to invoke the selected web services.

V. RESULT OF THE PROPOSED WORK

In fig-3 all the agents are running to work with our architecture after first initialization. Each and every agent has their unique roles.The following figure-4 has shown the communications among the agents to discover the services using our framework. To complete our job we have chosen the JADE technology. JADE is a very powerful middleware framework built with Java to design a MAS based architecture. It also come with many interactive inbuilt tools which are very useful during the development of MAS based system. One very important add-on that we have used is the Web Service Dynamic Client (WSDC)[13][14] in our propose system. The Web Service Invocation component selects the client application to invoke a particular Web Service and these clients are developed using the WSDC. One encouraging point about the WSDC is that, to program a client system using WSDC add-on, developers don't need to know about JADE's internal architecture, event they don't need to any details about agent's life-cycle.

VI. CONCLUSION AND FUTURE WORK

In our previous works we have tried to develop some new frameworks for better automaticity, with a better web service composition, discovery and selection. We didn't develop any new Semantic Execution Environment. Rather, our work was heavily dependent on the existing works like IRS-III, WSMX etc. In this paper we propose a framework for SWS discovery that communicates between multi agent system and web services. In those previous works use of agent based technology was very rare. This time we totally turn our vision to Multiagent based technology (JADE) to develop a new framework. We have introduced some agents who are eligible to perform the job of semantic matching between goal and web service description has been included thus removing the requirements of any other Semantic Execution Environment.

In this work security aspect of MAS System is not considered here. So it will be the important concern of our future works.

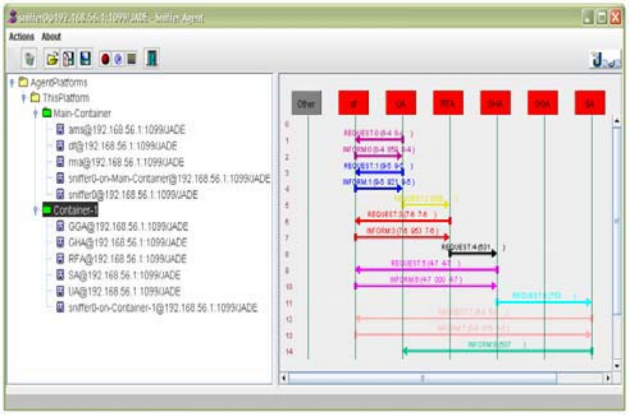

sniffer0@192.168.56.1:1099/JADE - Sniffer Agent
Actions
About
AgentPlatforms
ThisPlatform
Main-Container
ams@192.168.56.1:1099/JADE
df@192.168.56.1:1099/JADE
rma@192.168.56.1:1099/JADE
sniffer0-on-Main-Container@192.168.56.1:1099/JADE
sniffer0@192.168.56.1:1099/JADE
Container-1
GGA@192.168.56.1:1099/JADE
GHA@192.168.56.1:1099/JADE
RFA@192.168.56.1:1099/JADE
SA@192.168.56.1:1099/JADE
UA@192.168.56.1:1099/JADE
sniffer0-on-Container-1@192.168.56.1:1099/JADE
Other
df
UA
RFA
GHA
GGA
SA
REQUEST:0
INFORM:0
REQUEST:1
INFORM:1
REQUEST:3
INFORM:3
REQUEST:4
REQUEST:5
INFORM:5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Fig.4. Result of our proposed architecture that showing how the information get shared between the agent in the said framework

REFERENCES

[1] E. Cerami. "Web Services Essentials", Publisher: O'Reilly,ISBN: 0-596-00224-6.

[2] V. Vasudevan, "Web Service Primer", http://www.xml.com/pub/a/ws/2001/04/04/webservices/index.html

[3] M. Kalin, "Java Web Services: Up and Running", 1st Edition,Publisher: O'Reilly Media, Inc.,ISBN-13: 978-0-596-52112-7

[4] Web Service Description Language (WSDL) 1.1, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[5] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[6] UDDI Version 3.0.2, http://uddi.org/pubs/uddi_v3.htm

[7] JADE website, http://jade.tilab.com/

[8] D. Grimshaw, "JADE Administration Tutorial", http://jade.tilab.com/doc/tutorials/JADEAdmin/index.html

[9] G. Caire, "JADE Programming Tutorial for Biginners",

[10] http://jade.tilab.com/doc/tutorials/JADEAdmin/index.html

[11] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa and R. Mungenast, “JADE Administrator’s Guide”, http://jade.tilab.com/doc/tutorials/JADEAdmin/index.html

[12] F. Bellifemine, G. Caire, T. Truccoand G. Rimassa, “JADE Programmer’s Guide”, http://jade.tilab.com/doc/programmersguide.pdf

[13] JADE API Documentation; http://jade.tilab.com/doc/api/index.html

[14] Web Service Dynamic Client add-on, http://jade.tilab.com/dl.php?file=DynamicClientAddOn-2.0.zip

[15] E. Scagliotti and G. Caire, “Web Services Dynamic Client Guide”, http://jade.tilab.com/doc/tutorials/DynamicClientGuide.pdf

[16] A. G. Neiat, r. Forsati, M. Mohsenzadeh and A. M. Rahmani, “An Agent-based Semantic Web Service Discovery Framework”, IEEE Computer Society, DOI 10.1109/ICCMS.2009.75,

[17] G. Hong-jie, M. Fan-rong, S. Jin-fei and D. Peijun, “Web Service Discovery Based on the Cooperation of UDDI and DF”, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04680816

[18] S. Iftikhar, H. F. Ahmad and H. Suguri, “Dynamic Interoperability of Software Agents with Semantic Web Languages”, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04600264

[19] D. Greenwood and M. Calisti, “Engineering Web Service – Agent Integration”, http://www.whitestein.com/library/WhitesteinTechnologies_Paper_IEEESMC2004.pdf

[20] WSIG Guide, http://jade.tilab.com/doc/tutorials/WSIG_Guide.pdf

[21] WSIG add-on, http://jade.tilab.com/dl.php?file=wsigAddOn-2.3.zip

[22] K. Sycara, M. Paolucci, J. Soudry and N. Srinivasan, “Dynamic Discovery and Coordination of Agent-Based Semantic Web Services”