Information Retrieval Multi-Agent System Using Web Services

Yu Mon Zaw[#1], Nay Min Tun[*2]

[#] *Information System Department, Computer University (Tha-Hton),*
Mon State, Myanmar
[1] mayumonzaw@gmail.com

[*]*Computer University (Kyaing Ton),*
Shan State, Myanmar
[2]naymin.300777@gmail.com

***Abstract*— Information Retrieval Systems are playing in important role for every Internet user since the World-Wide-Web has started flourishing with enormous information resources of different sectors and trends that can give public knowledge and know-how. As the current industrial and business trend is not only providing knowledge but also supporting service, Web Service technology becomes popular because it is standards-based realization of the service-oriented computing (SOC) paradigm, which has emerged in response to a fundamental shift in the way enterprises conduct their business. Retrieving desired specific information and facts on the Web cannot be done by single Web Service. So there should be a possibility to combine existing related Web Services in order to fulfill user requests. To compose Web Services, Multi-Agent System can give a great help. In order to construct a Multi-Agent System from a collection of Web Services, we propose an Agent Algorithm and a Web Services Based IR framework. The implementation and testing of this system utilizes the real datasets of clinics in the Yangon area.**

***Keywords*— Multi-Agent, Web Services, Service Oriented Computing, Information Retrieval, Web**

I. Introduction

The amount of information that is available on the Web these days can only be called enormous, and it keeps growing daily. Searching desired information among such enormous resources becomes time consuming. As World Wide Web has gradually transformed into business and service trend, Web services are a new and promising technology which standardizes many aspects of distributed processing and communication on the World-Wide-Web. A Web service is an accessible application that other applications and humans can discover and invoke to satisfy multiple needs. Web Services can support many companies and associations in order to promote the services and facilities they provide such as banking system, travel tour agencies and so on.

However, amount of products and services available now on the Web increases dramatically and goes beyond user's ability to analyze them efficiently. At the same time the number of potential customers available via the Internet also increases significantly and starts to be beyond service providers' ability to perform efficient targeted marketing. In particular, if no single Web Service can satisfy the functionality required by a user, there should be a possibility to combine existing services together in order to fulfill the request [3].

At the same time, multi-agent systems (MAS) represent another distributed computing paradigm based on multiple interacting agents that are capable of intelligent behavior. Multi-agent systems are often used to solve problems by using a decentralized approach where several agents contribute to the solution by cooperating one each other. One key feature of software agents is the intelligence that can be embodied into them according to some collective artificial intelligence approach that needs cooperation among several agents that can run on a parallel or distributed computer to achieve the needed high performance for solving large complex problems keeping execution time low [1].

Therefore, for the Web-Wide Information Retrieval system based on Web Services, it is sure that not a single Web Service can fulfill the user needs. To get the complete and desired information results, numerous domain related Web Services should be cooperated. In this case, we propose to get the help of multi- agents systems.

The remainder of this paper is organized as follows. In the next section, we will introduce the background knowledge and theory of Web Services and Agent Computing. Section III describes the related work. Section IV discusses about our proposed framework. We introduce our system components, their functions and natures in Section V. Section VI presents proposed Algorithm for our Medical IR Multi-Agent System. Section VII describes selecting the most appropriate web method. Implementation and Testing of the proposed system is presented in Section VIII. Evaluation of system performance states in Section IX. The paper concludes in Section X with titled Conclusions.

II. BACKGROUND

A. Web Service

A Web Service[4] is an accessible application that other applications and humans as well, can automatically discover and invoke. An application is a Web Service if it is

- independent as much as possible from specific platforms and computing paradigms;
- developed mainly for inter organizational situations rather than for intra-organizational situations; and
- easily composable (i.e., its composition with other Web Services does not require the development of complex adapters).Web Services are, in practice, transient and stateless processes that exist only during service execution, which is triggered by a request coming from a consumer, or client. Services are instantiated to perform specific tasks, thus facilitating scalable, concurrent service provision. The design of a Web Service is usually defined as a clearly articulated workflow, for the sake of reliability and quality of service.

Though Web Services has many advantages, but still there are certain problems which need to be addressed. These are:

- Provided resources and services are not in machine understandable form, these are in human understandable form.
- The representation of resources and services on the web are unstructured and they are loosely related to each other.
- Searching resources and services on the web at present is keyword based; no semantics of the resources are used. So by using some popular keywords, web page owner can make his page mostly retrieval with irrelevant results and
- Interoperability between toolkits.

The Web services framework intends to provide a standards-based realization of the service-oriented computing paradigm, which has emerged in response to a fundamental shift in the way enterprises conduct their business. Fully integrated enterprises are being replaced by business networks in which each participant provides the others with specialized services. Traditional IT infrastructures in which infrastructure and applications were managed and owned by one enterprise are giving way to networks of applications owned and managed by many business partners. Standards and the pervasiveness of network technologies provide the technology support for this trend.

This new computing environment defines a set of requirements that distinguish SOC from other computing paradigms. To operate in a SOC environment, applications (“services”) must declaratively define their functional and nonfunctional requirements and capabilities in an agreed, machine-readable format. Based on declarative service descriptions, automated service discovery, selection and binding become a native capability of SOC middleware and applications. A consequence of the dynamic binding capability is a looser coupling model between applications.

B. Agent Computing

An agent [1] is a computational entity that acts on behalf of another entity (or entities) to perform a task or achieve a given goal. Agent systems are self-contained software programs embodying domain knowledge and having ability to behave with a specific degree of independence to carry out actions needed to achieve specified goals. They are designed to operate in a dynamically changing environment**.**

Agents typically include a set of features. The main features of agents include the following:

- *Autonomy:* the capacity to act autonomously to some degree on behalf of users or other programs also by modifying the way in which they achieve their objectives.
- *Pro-activity:* the capacity to pursue their own individual set goals, including by making decisions as result of internal decisions.
- *Re-activity:* the capacity to react to external events and stimuli and consequently adapt their behavior and make decisions to carry out their tasks.
- C*ommunication and Cooperation*: the capacity to interact and communicate with other agents (in multiple agent systems), to exchange information, receive instructions and give responses and cooperate to fulfill their own goals.
- *Negotiation:* the capability to carry out organized conversations to achieve a degree of cooperation with other agents.
- *Learning:* the ability to improve performance and decision making over time when interacting with the external environment

III. RELATED WORK

Yue-San Chang, Chao-Tung Yang and Yu-Cheng Luo presented an Ontology based Agent Generation for Information Retrieval on Cloud Environment. [2] While user submitting a flat-text based request for retrieving information on a based on predefined ontology and reasoning rule, and then be translated to a Mobile Information Retrieving Agent Description File (MIRADF) that is formatted in a proposed Mobile Agent Description Language (MADF). A generating agent, named MIRA-GA, is also implemented to generate a MIRA in accordance with MIRADF.

Andrea Addis, Giuliano Armano, Francesco Mascia, and Eloisa Vargiu from University of Cagliari, Italy proposed a News Retrieval through a MultiAgent System [5]. In this paper, a multiagent system devised to generate press reviews has been presented. The system encompasses three main tasks. First is extracting articles from online newspapers. Second task is classifying them using hierarchical text categorization. The final task is providing suitable feedback mechanisms.

Vishal Jain proposed the information retrieval practical model through the multi-agent system with data mining in a cloud computing environment. [6] He recommended that users should ensure that the request made to the IaaS is within the scope of integrated data warehouse and is clear and simple. In that research model/ architecture, the use of cloud computing allows the users to retrieve meaningful information from virtually integrated data warehouse that reduces the costs of infrastructure and storage.

Our proposed system applies Web Services in order to retrieve information instead of mobile agents. By the use of web services, it does not need to know the each Machine's database detailed schema and does not need to handle the whole database so that it can solve database security issue. By the help of agents, desired information can be searched in a set of Web Services instead of single Web Service so that searching will be more effective at one sitting. Moreover, decision making on which web methods to be called can also be solved by agents' knowledge so that system will be more intelligent.

IV. The Proposed System Framework

We propose a framework for Web Services Based Information Retrieval Agent System running on World-Wide-Web. The proposed system framework is intended to apply in Medical field. In that environment, a number of hospitals, clinics and health care services are hosted and are providing Web Services. Each Web Service of a specific hospital offers specialists (doctors) information worked at that hospital by numerous Web Methods. By using our Medical IR multi-agent System, users (patients) can easily search the desired information by day (Monday, Tuesday,...), by time (1pm-4pm,…), by doctor's name (Prof: Dr. Nay Win,…), by specific clinic (Asia Royal, SSC,..) and by disease type (Liver, Lung, OG,…).

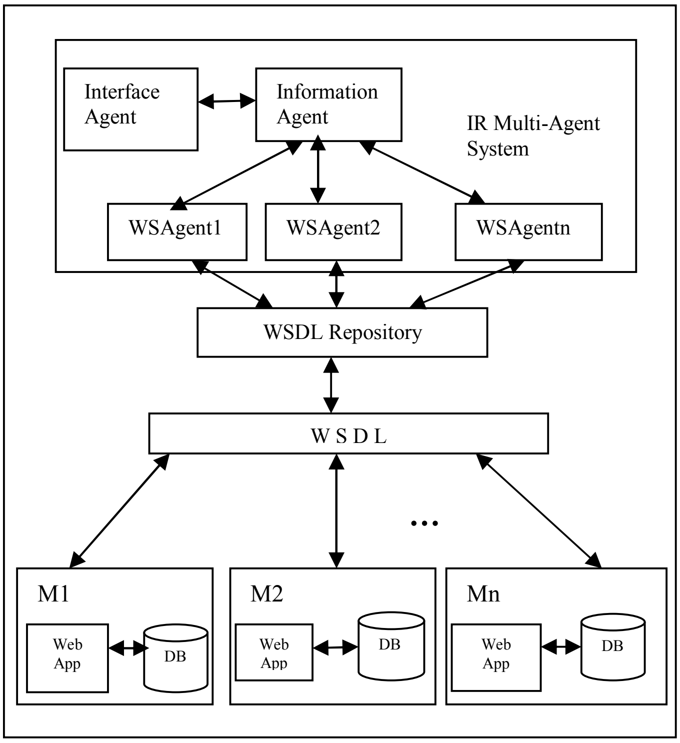

Fig. 1 Medical IR Multi-Agent System Architecture

In Fig.1, the proposed Medical IR Multi-Agent System Architecture is described. The whole agent platform is running on main server machine. Other Web Services providers are resided on different machines. The detail explanation about Fig. 1 and all components' detail functions are stated in section V.

V. System Components and Their Functions

A. *Interface Agent*

The system will start from Interface Agent. Receiving queries from end users to search required information from distributed databases and showing back the queries result may be done by interface agent in this proposed system. The duty of Interface Agent in this system is to receive queries from end users, prepare the queries into a format match for Information Agent's working style, pass the well formatted data to Information Agent and to show back the queries results.

B. *Information Agent*

Information Agent is the main agent who distributes user input query to a set of WSAgents depending on conditions and then combines the returned result of WSAgents and sends the final result to Interface Agent.

C. *WS Agents*

There are a set of WSAgents which invoke specific web service associated with them. One WS Agent has to handle one WSDL. In that case, one WSDL may contain more than one web methods. The duty of WS Agents is to make the best choice in selecting the web method according to the data parameters sent by Information Agent. They will send back a message to Information Agent whether they get the desired data or not.

D. *WSDL Repository*

WSDL Repository stores Web Service Description files published by various web applications hosted on the Web. In our system, to make comfortable for WS Agents in sending message and carrying data, we generate each WSDL into a set of client classes and so WS Agents can handle the retrieved results in object and can carry inside the messages.

E. *A set of Machines (Ms)*

Several web application systems from Ms (Machines) support services for Information Retrieval purpose**.** One M represents one hospital. Each M contains a web application (dynamic web projects) for each hospital which desires to coordinate with our Medical IR Multi-agent System. They support Web Service with many web methods. They possess databases with different schemas. Each web method contains SQL statement to access the database according to the received parameter values.

VI. Proposed Algorithm for Medical IR Multi-Agent System

- User Request is defined as Req.
- Refined Request is defined as RefineReq.
- Final Result is defined as FinResult.
- The Returned Result from each Web Service is defined as ResWS.
- ResWS1, ResWS2,…, ResWSn ϵ ResWS

- WS is the set of Web Services published in WSDL Repository.
- WS1, WS2,…, WSn ϵ WS
- AG is the set of all agents in the System.
- InterfaceAG, InfoAG,WSAG1,WSAG2,…,WSAGn ϵ AG
- Interface Agent is defined as InterfaceAG.
- Information Agent is defined as InfoAG.
- Web Service Agent is defined as WSAG.

```
Begin
FinResult←NULL.
SelectedWSAG ←NULL.
User sends Req to InterfaceAG.
RefineReq←Req refined by InterfaceAG.
InterfaceAG sends RefineReq to InfoAG.
SelectedWSAG ←InfoAG determines which WSAGs to be called.
While not receiving the FinResult from InfoAG
    InfoAG sends RefineReq to WSAG1, WSAG2, …, WSAGn ϵSelectedWSAG.
    For All WSAG1, WSAG2, …,WSAGi…, WSAGn
        WSAGi determine which web method of WSi to invoke according to the received RefineReq parameters.
        If WSAGi can solve RefineReq
            It will return ResWSi to InfoAG.
        Else
            Return NULL.
        End If
        FinResult←FinResult+ ResWSi.
    End For
End While
End
```

VII. SELECTING THE MOST APPROPRIATE WEB METHOD

Selecting the most suitable web method is the special duty of Web Service Agents (WSAGs). There will be more than one web methods are resided in one WSDL with various parameter types and values. For example, WSDL of Hospital1 possesses searchByDoctorName(String DoctorName) method. The return type is Class Hospital1DoctorSchedule Array. Some part of Hospital1 WSDL related with this web method is shown as below.

```
<element name="searchByDoctorName">
  <complexType>
   <sequence>
    <element name="docName" type="xsd:string"/>
   </sequence>
  </complexType>
 </element>
 <element name="searchByDoctorNameResponse">
  <complexType>
   <sequence>
    <element maxOccurs="unbounded"
name="searchByDoctorNameReturn"
type="tns1:Hospital1DoctorSchedule"/>
   </sequence>
  </complexType>
 </element>
```

To accomplish it, every WSAG requires three main components. The first is Input Data Format File (IDFF) to learn the input parameter contained in the message sent by InformationAgent. The second is Web Methods Description File (WMDF) which states what web methods possess how much parameter passing values, describes what that values type are (String, Integer, etc),describes the return type and states their purpose. The mappings between IDFF and WMDF are described in Mapping Parameter and Methods File (MPMF) in which what web methods should be called according to the received parameter values are stated.

In IDFF, there are 5 input field: DoctorName(String), Disease(String), Day(String), StartTime(Integer), EndTime(Integer). These 5 input type format and received message parameters have to perform miss or match value checking. If the received message contains only one parameter that is for DoctorName field, the related web method must be with one String parameter passing type. So, according to the predefined rules and knowledge in MPMF, WSAG decides the method "searchByDoctorName(String DoctorName) to be called.

VIII. IMPLEMENTATION AND TESTING

We implemented this proposed system based on J2EE and JADE platforms. We used apache tomcat server, axis for web development and MySQL database. The system has run on 6 Intel Core 2 Duo machines and host OS are Ubuntu 12.04 LTS. One machine is for the main Multi-Agent System running environment and others are holding Web Services.

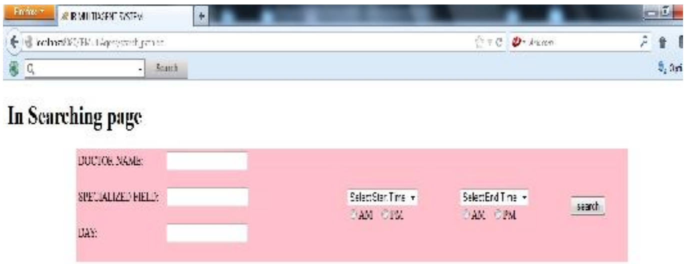

Fig.3 Information Search Page

Fig. 3 is the sample form of Search Page. Users can find their desired doctors' schedule by name, by doctor's specialized field (eg. Cardiologist), by Day (eg. Tuesday,..), by Time (eg. 11 am to 8 pm). Users can search by using at least one criteria or all .The example searching result will be in the form of Table 1.

TABLE I
SAMPLE SEARCHING RESULT

Name	Specialist Field	Hospital	Day	From	To
Prof. Dr. Cho Lay Mar	Cardiologist	Bahosi	Tuesday	1 PM	3PM
Dr. Ni Ni Win	Cardiologist	Asia Royal	Tuesday	5PM	7 PM

IX. EVALUATING PERFORMANCE RESULT

In the proposed IR Multi-Agent system, there are three main agent types: Interface Agent, Information Agent and Web Service Agents. The number of Web Service Agents depends on the number of WSDLs they must handle in the system. So, the more WSDLs exist, the more WSAgents the system require and the more complex the system will be. Therefore, more time will be consuming. But according to our testing, the processing time difference between increasing WSAgents usage is quite small and is acceptable.

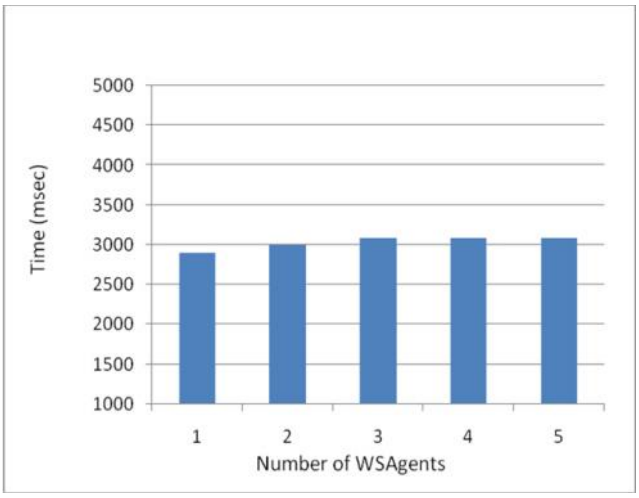

Fig.4 Comparison for Average Processing Time on increasing WSAgents

In Fig.4, the horizontal row in graph represents number of WSAgents and the vertical column represents time(mili seconds) taken by the system. According to the comparison graph, the processing time has increased with small difference and the increasing rate is stable when the system is tested with 3, 4 and 5 WSAgents.

X. CONCLUSIONS

Information Retrieval Multi-Agent System using Web Services is proposed. Efficiently composed Web Services using multi-agents features can give new form for Web wide information retrieval systems. It also states new form of web methods selection procedure in order to make retrieving information easily. The proposed system will become an intelligent way for searching or retrieving information from World Wide Web. By implementing the propose system, it can give a good hand for the public to get the desired specialists' schedule completely and perfectly at one sitting and can make the right choice with their current situations. Moreover, this framework can be applied in other domain area efficiently. This proposed system has been intended to extend into cloud wide information searching and retrieving model.

REFERENCES

[1] D. Talia, *"Cloud Computing and Software Agents : Towards Cloud Intelligent Services"*.

[2] Y. S. Chang,C. T. Yang and Y. C. Luo, *"An ontology based agent generation for information retrieval on cloud environment"*, Journal of Universal Computer Science, vol. 17, no. 8, 2011.

[3] G. Vadivelou, E.Ilavarasan and M.S. Yasmeen, *"QoS based semi-automatic web service composition using multi-agents systems*", IJCSET, Vol 1, Issue 7, pp. 381-386, August 2011.

[4] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana, *"The Next Step in Web Services"*, Comm. ACM, vol. 46, no. 10, Oct. 2003.

[5] A. Addis, G. Armano, F. Mascia and E. Vargiu, "*News Retrieval through a MultiAgent System"*, Web Offset Association Conference (WOA), 2007.

[6] V. Jain, *"Information retrieval through multi-agent system with data mining in cloud computing*", J.Comp.Tech.Appl, Vol 3 (1), pp. 62-66, IJCTA, 2012.