A Multi-agent Based Framework for Dynamic Service Discovery and Access Using Matchmaking Approach

T.Amudha
Lecturer, Bharathiar University
Coimbatore – 46
amudha.swamynathan@gmail.com

N.Saritha
Research Scholar, Bharathiar University
Coimbatore – 46
sarithanallasamy@gmail.com

P.Usha
Research Scholar, Bharathiar University
Coimbatore – 46
usha5310@gmail.com

***Abstract*— Service-oriented architecture (SOA) has gained recognition due to its reusable nature of applications and interoperability between heterogeneous technologies. The basis of SOA is the service whereas service is a self-contained software module that performs a predetermined task. The most important tasks in building a service-oriented application are the creation of services with coarse-grained functionality and the creation of interfaces with the right level of abstraction that enhances service discovery and access. Dynamic Service Discovery is one of the key issues of SOA that requires new avenues of research. Agents are reusable s/w components that provide controlled/shared access to services and resources and forms the building blocks with collaborators called as collaborative agents for applications organized across various networks. This paper proposes a novel approach for a multi-agent based dynamic service discovery using matchmaking technique.**

Keywords- Multi-agent system, Service discovery, Matchmaking, Middle-agent.

I. Introduction

Numerous services that provide shared access features are published in the internet. The service information is maintained in the local storage of the provider and it is found to be a complex task to discover the required service. Most of the service discovery methods that are available to search the services are implemented using domain based and query based techniques [6]. Such methods seem to be time-consuming and require constant user communication and consultation.

Intelligent agents provide an ideal mechanism for building and integrating systems that meet a wide variety of requirements [5]. Building sophisticated agents need specialized skills and knowledge in a variety of areas including agent architecture, communications technology, reasoning systems, knowledge representation, agent communication languages and protocols. Agent-based computing is an approach that facilitates the design and development of sophisticated systems by viewing the system as a society of independent communicating agents working together to meet the goals. Agents can have in-built domain expertise, they can learn, reason logically and adapt to new circumstances. Agent-based systems are constructed from a loosely coupled set of software components. Middle-agents support the flow of information in Multi-Agent Systems (MAS), assisting in locating and connecting the ultimate service provider with the service requester [2]. Middle agents deal with both preferences information and capabilities information, and are neither requesters nor providers. Finding an appropriate provider for a requester through a middle agent is termed as Matchmaking [1,7,8]. Service Provider agents advertise their capabilities via middle agents which store these advertisements. Service requester agents check with the middle agent for a desired service, and the middle agent matches the request against the advertisements and returns the result. This paper provides a flexible, robust multi-agent based framework that is designed to provide an efficient service discovery mechanism to obtain the required services from a collection of shared services.

978-1-4244-4711-4/09/$25.00 ©2009 IEEE

II. RELATED WORKS

SOA provides a clear solution to the application integration issues by allowing systems to expose their functionality via standardized, interoperable interfaces (Panda,2008). Their work also stated several key advantages of SOA such as i)Adaptation of applications to changing technologies
ii)Easy integration of applications with other systems iii)Leveraging of existing investments in legacy applications and iv)Quick and easy creation of a business process from existing services.

In their work, Palathingal and Chandra (2005) suggested an Agent Approach for Service Discovery and Utilization (AASDU) that focuses on using lightweight autonomous agents, built into a multi-agent community called the Multi Agent Referral System and Web Service standards namely UDDI, SOAP, WSDL and XML. AASDU is a query-based approach where agents interact with the end user to discover services, to specific queries, and efficiently manage their utilization.

Ongoing research works in the area of Web Services comprises of service providers developing and publishing specialized services with private or public registries depending on its usage (Sheng-Tzong Cheng et al., 2002). To reveal the utility of Semantic Web Service descriptions for service composition, a goal-oriented, interactive composition technique using matchmaking algorithms was developed by Evren Sirin, Bijan Parsia, and James Hendler, 2004, to help users filter and select services while building the composition.

III. SYSTEM ARCHITECTURE

The middle-agent based dynamic service discovery system is intended to make an interaction between the service publisher and service subscriber through the middle agents under multi agent environment.

The system comprises of the following three components for service discovery and access viz., the shared service space, the service discovery manager and the client.

A. *Shared Service Space*

The service publishing and searching operations are done with the Shared Space, which maintains the services with its description. This component takes responsibility of the services published in the local distributed environment. In a distributed service environment, multiple virtual shared spaces are used. It receives the services and service details from the publisher. Description of all the services published is updated into the shared service space. The service details are distributed to the subscribers through the service discovery manager component.

B. *Service Discovery Manager*

The discovery manager component deals with subscriber management, service discovery and service access requirements. The subscriber can access any service provided by the service publisher through the service discovery manager. The service discovery process is facilitated with domain and description information to assist the client in finding out the relevant services with reference to the client requirements. The matchmaking technique is applied to detect similar services under the publisher environment. This component acts as the centralized authority for the entire service discovery framework

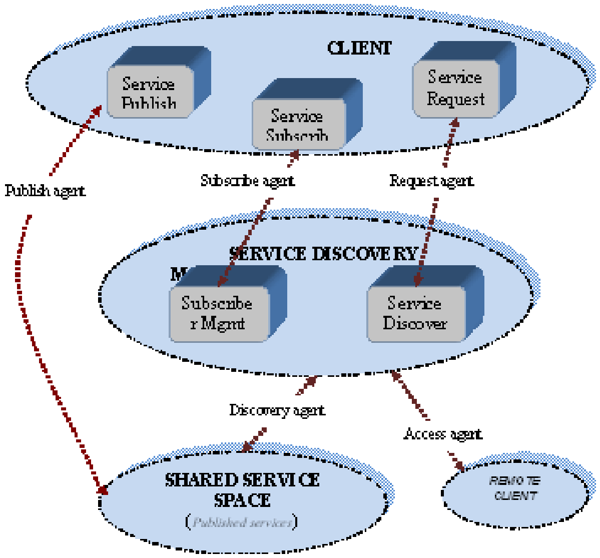

Fig 1**:** Proposed Architecture for Dynamic Service Discovery

C. *Client*

The client component manages with service-publish, service-subscribe and service-request tasks initiated by the clients. The publisher and the subscriber are the two major roles executed by the clients. The client publishes the services directly on to the shared service space. Client may subscribe or request for any specific service via the service discovery manager, which locates and provides the necessary services to the client.

IV SYSTEM IMPLEMENTATION

The match making system using middle agents is designed to provide various services for the subscribers. The discovery manager is the service search authority for the system. The operations under the shared space, discovery manager and the client applications are carried out with the support of the multi-agents. The middle agent classifies services as either matchmaking services or brokering services. Matchmaking services are passive services whose goal is to provide a client agent with a list of names whose properties match its supplied criteria. The agent may later contact the matched agents to request services. On the other hand, brokering services are active services that directly deliver a message to the relevant agents on their clients' behalf.

In both types of services, an agent advertises itself by sending a message which contains its name and a description of its characteristics to a middle agent. There are two types of information used in the agent interaction process: preferences and capabilities. To protect the privacy of each agent, preferences flow from a service requester agent to a service provider agent, and capability information flows from the service provider agent to the service requester agent. Agents that deal with both types of information that are neither requesters nor providers are termed as middle-agents. Matchmaking is the process of finding an appropriate provider for a requester through a middle agent. Service Provider agents advertise their capabilities via middle agents who store these advertisements. Whenever the requester checks with the middle agent for a desired service, the middle agent matches the request against the advertisements and returns the result.

With the target of building a cross-platform compatible system, the dynamic service discovery system was developed using JAVA. Datagram was used for inter-agent communication ensuring minimum bandwidth and overhead, using features of the JAVA environment to ensure low maintenance costs and implemented machine learning based intelligence on the agents to optimize searching and to improve the response time. XML is used to set up the data values used under the distributed environment. The shared information is composed as XML contents and is distributed to the agents. Given are the agents developed to implement service discovery

A. Publish Agent

The publish agent is the initial agent for the system that handles service publish operations. The publisher client initiates this agent. The service and publisher information are embedded with this agent. This agent transfers the service publish details to the shared space environment. The publish information are prepared as XML documents using the XML compose agent. The XML parse agent is used to parse and analyze the XML data values.

B. Subscribe Agent

The subscribe agent is used to subscribe the services under the discovery manager environment. The discovery manager maintains the subscription list and provides the service information to the subscribers with reference to service domain details. The client generates the subscribe agent and forward it to the discovery manager. The subscriber details are updated during the service subscription process and the service information are distributed with respect to the subscription data.

C. Request Agent

The request agent is used to invoke the services. The request is passed to the shared space to access the relevant services. The request agent is generated at the client location, transferred to the discovery manager to identify the required service and finally executed at virtual space environment. Service fetching is the main task for the request agent.

D. Discovery Agent

The discovery agent is the vital middle agent for the system. It is connected with the client agent and shared space applications. The discovery agent matches the service request from the client with the dynamically hosted services and enables fetching of corresponding services.

E. Access Agent

The access agent is designed to access the service at the remote client. The service access is initiated by the request agent and the services are fetched from the service providers through the discovery manager. The service access agent can be initiated from any client, who may be a subscriber or a general service requester.

F. XML Compose Agent

The XML compose agent handles the XML document preparation tasks. The tag name indicates the name of the data values. All the agent-to-agent communication is done using the XML format. The sender agent uses the XML compose agent and the receiver agent uses the XML parse agent to analyze the information.

G. XML Parse Agent

The XML parse agent is designed to parse and analyze the received information. The XML documents are passed into XML parse agent for data value extraction process. The extracted values are assigned to the relevant field elements. The XML compose agent and XML parse agent are integrated in all the other agents

V Discussions

The system is designed with the prime motive to handle concurrent submission and dynamic discovery of services by the middle-agents without user intervention. The matchmaking technique is further improved with dynamic service hosting and searching tasks with unlimited number of clients. Filtering techniques such as domain based filtering and description based filtering are used to make the service discovery process more efficient. It incorporates XML for the distribution of service information among the components and dynamically composes and parses XML documents for the data distribution process. Since all the communications between agents are in the XML format, datagram based messaging proved effective. The platform independent JAVA environment provides the system with a uniform application environment protecting the agent from variations in OS implementations and network architecture.

VI CONCLUSIONS AND FUTURE WORK

The dynamic nature of the networked environment demands that any solution for locating services be able to inherently deal with the unpredictable nature of its operating environment. The multi-agent based framework for dynamic service discovery suggests a remedy in the form of independent agents traversing to discover and access services with complete autonomy of action.

This research work clearly proves that unlike the commonly used solutions for achieving improved search throughputs such as indexing and caching, this solution achieves even higher performance with much lower communication and processing overheads. The solution can demonstrate that by using the suitable paradigm to solve the problem, in this context, Intelligent Agents, a dramatic increase in solution efficiency can be achieved. A learning based artificial intelligence (AI) is used to increase the efficiency of the solution. The AI analyses the performance of the attached search agent and derives ways to make future searches more efficient.

Agent construction toolkits allow software developers with problem domain expertise to easily build software agents. In this paper we proposed a multi-agent based architecture for efficient service discovery and access and we conclude that this architecture shall be built using agent toolkits such as JADE, Grasshopper, ABLE etc., to achieve promising results.

REFERENCES

[1] Daniel Veit, Jörg P. Müller, Christof Weinhardt,: Multidimensional Matchmaking for Electronic Markets, Submitted for publication in the Applied Artifical Intelligence Journal,(2002).

[2] Decker, K., K. Sycara, and M. Williamson.: Middle-agents for the internet. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), 578-583. (1997).

[3] Evren Sirin, Bijan Parsia, and James Hendler,: Filtering and Selecting Semantic web Services with Interactive Composition Techniques, IEEE Intelligent systems, 1541-1672, (2004).

[4] Huhns M. N, : Agents as Web services, Internet Computing, IEEE,93-95, 6 (4), (2002).

[5] Huhns, M. N. et al: Research Directions for Service-Oriented Multiagent Systems. IEEE Internet Computing, 9 (6) (2005)

[6] Paul Palathingal, Sandeep Chandra,: Agent Approach for Service Discovery and Utilization, White Paper, (2005)

[7] Sycara, K., Widoff, S., Klusch, M., Lu, J.: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. ,Autonomous Agents and Multi-Agent Systems 5, 173-203, (2002).

[8] Veit, D., J. P. Müller, M. Schneider, B. Fiehn.:Matchmaking for Autonomous Agents in Electronic Marketplaces. In Proceedings of the 5th Intl. Conference on Autonomous Agents, Montreal, 65-66, (2001).