SACoSS-semantic Agent Based System for Cloud Service Suggestion Using Cloud Service Ontology

B.Saravana Balaji\textsuperscript{1,*}, T.N.Prabhu\textsuperscript{2}, Dr.N.K.Karthikeyan\textsuperscript{3}

\textsuperscript{1}Research Scholar, Anna University of Technology, Coimbatore, India
\textsuperscript{2}Assistant Professor-IT, Sri Ramakrishna Engineering College, Coimbatore, India
\textsuperscript{3}Professor and Head-IT, Sri Krishna College of Engineering and Technology, Coimbatore, India

Abstract

With increasing demand and adoption of cloud services through web, selecting an appropriate cloud service according to requirements of the consumer becomes a great issue. In this paper we present a system for providing a list of cloud services as suggestion according to the consumer requirement, called SACoSS which is built using semantic technologies. SACoSS divides the cloud services in two domains based on SLA and WSDL of cloud services as SaaS services and IaaS services and updates the ontologies accordingly. The workflow of SACoSS system is of two steps, cloud service ontology updation and cloud service suggestion generation. The performance evaluation of SACoSS shows that this system provides better results and achieves higher prediction accuracy than other methods.

Keywords: Cloud Computing; Ontology; SLA; Semantic Agents; Service Selection

1 Introduction

The emergence of Grid Computing and Service Oriented Architecture makes a new computing paradigm called cloud computing. The main idea behind the emergence of cloud computing is to reduce the cost of computing and also to increase the reliability, availability and flexibility [8]. Cloud computing is provided in three formats called Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). The main players of cloud computing are cloud service providers and cloud service consumers. In cloud computing, the requirement of service consumer is to have the minimum hardware and software, and need only pay to service providers to get the required compute resources. Under the diverse nature of cloud computing, a cloud service consumer faces numerous technical issues on selecting a particular cloud service like, the consumer must understand the technical requirements, and workflow of the service etc. Normally, these kinds of information about a particular cloud service are provided through Service Level Agreements (SLA) and WSDL documents by the cloud service provider. SLAs contains information like trading parties, attributes, measurement metrics, and attribute values.
etc [19]. Regardless of availability of SLAs and WSDL documents the cloud service consumer faces the problem of selecting the appropriate service according to the need as the SLAs describe the services through non-standardized attributes and WSDL document describes service technical attributes [18]. To solve this issue, semantic web technologies could be used.

In the Semantic Web, knowledge is represented by means of ontologies [3], which are viewed as a formal specification of a domain knowledge conceptualization. The semantic descriptions of services in the form of ontologies have emerged due to their ability for supporting the automation of service retrieval, invocation, composition and monitoring tasks by providing machine-exploitable meaningful declarative descriptions of service characteristics [5]. In this paper, we propose SACoSS-Semantic Agent based System for Cloud Service Suggestion using Cloud Service Ontology, which uses the SLA and WSDL document of cloud services to extract the knowledge about the cloud service and answers the cloud service requester (consumer) with appropriate cloud services as suggestion according to the requirements of the requester (consumer).

The remainder of this paper is organized as follows. Section II describes various related methods proposed previously. Section III explains about the architectural components of the SACoSS. Section IV provides information about workflow of SACoSS. Section V gives details about the performance evaluation of the SACoSS by comparing with few existing methods under different metrics. Section VI consists of conclusion and future work that can be extended from the SACoSS.

2 Related Work

The problem of selecting an appropriate service involves multicriteria decision making situation and there has been several methods were proposed by many researchers around the globe. The method proposed in [4] addresses Automated web-based negotiation of Service Level Agreements (SLA) which can help define the QoS requirements of critical service-based processes, is achieved through a novel trusted Negotiation Broker (NB) framework and by defining mathematical models. Also, intelligent selection algorithm is being used for adopting decision functions. A method for the construction of ontology automatically is proposed in [23] by mining large-scale web resources such as eHow and wikiHow. This method attempts to build a huge situation knowledge base of human activities by means of text mining techniques that exploit the structure of the how-to descriptions, which is essential for context/situation-aware services.

A recommendation system for home network environment has been proposed in [11]. This method constructs ontologies for user profiles and home network service and by using those, a recommendation system has been developed. For resource selection in science cloud, a method has been proposed in [9]. This method describes an ontology-based representation of cloud computing environment which allows conceptualization of common attributes among cloud resources and describes relations among them semantically. A resource virtualization method using ontology is introduced and a new Virtual Ontology (VOn) is configured dynamically. For rapid and efficient merging of a number of ontologies, the Map Reduce model is adopted.

Web service selection based on context ontology and QoS has been proposed in [16]. It describes how automatic selection of best service provider done upon a request based on mixed context and QoS ontology for a given set of parameters of QoS. It supports an e-business framework and also emphasizes one dimensional quality of services. Discovering semantic web service using SPARQL and agents has been proposed in [15]. It describes how the SPARQL query language can be used
to express the preconditions and postconditions of services as well as the goals of agents. Here SPARQL query evaluation can be used to check the truth of a precondition in a given context, construct the postcondition that will result from the execution of a service in a context, and also determine whether a service execution with those results will satisfy the goals of an agent.

A service concept recommendation system for service ecosystem has been proposed in [6]. It describes the framework of service concept recommendation system, which is built upon a semantic similarity model. This method did ontology-based service matchmaking from the perspective of measuring the similarity of concept-concept, instead of measuring the similarity of query-concept utilized by those semantic matchmakers. In [24] a collaborative filtering approach for predicting QoS values of Web services and makes Web service recommendation by implementing a prototype called WSRec which helps to achieve better prediction accuracy has been proposed. In [21] an ontology based network management system has been proposed. It presents a system to collect information through the cooperation of intelligent agent software, in addition to providing warnings after analysis to monitor and predict some possible error indications among controlled objects in the network. This system could sketch the four main components of network management systems with the technique of graphic monitoring multi-agent, which effectively enhance and improve the network monitoring performance. A semantic based automated service discovery method has been proposed in [1]. It addresses the issue of Web Service discovery given non-explicit service description semantics that match a specific service request by semantic based service categorization and semantic enhancement of the service request and provides a solution for functional level service categorization. Latent Semantic Indexing (LSI) is utilized for efficient matching.

3 SACoSS - Architecture

SACoSS architecture is provided in figure 1. SACoSS consists of seven semantic agents and a cloud service ontology repository for suggesting appropriate cloud services.

Fig. 1: SACoSS - Architecture

The semantic agents of SACoSS system are

- SLA Processing Agent
• Semantic Annotating Agent
• Domain Specifying Agent
• SaaS Ontology Updating Agent
• IaaS Ontology Updating Agent
• Query Processing Agent
• Cloud Service Searching Agent

3.1 SLA processing agent

This agent is responsible for retrieving SLA and WSDL documents from various cloud services that are available on the web. Then this agent process the SLA and WSDL documents based on PoS tagging method defined in RASP system [2] through which the environmental template of the cloud service developed.

3.2 Semantic annotating agent

This agent takes input from SLA processing agent and adds annotations for the subjects or nouns present on the parsed documents. This annotation process is based on WSDL-S [12].

3.3 Domain specifying agent

This agent identifies the domain of the cloud service that is described in the environmental template of the cloud service. Depending on the domain of the cloud service the template will be forwarded to either SaaS ontology updating agent or IaaS ontology updating agent.

3.4 SaaS ontology updating agent

This agent updates the existing ontology for Software as a Service kind cloud services by adding classes and object values for that service via the information that is described in the environmental template and store it in the cloud service ontology repository. The ontology structure and processing method for updating ontology will be discussed in the next section.

3.5 IaaS ontology updating agent

This agent updates the existing ontology for Infrastructure as a Service kind cloud services by adding classes and object values, adding relationship between various resources of the service with the help of information that is described in the cloud service template and store it in the cloud service ontology repository. The ontology structure and processing method for updating ontology will be discussed in the next section.
3.6 Query processing agent

This agent will parse the query given by the user in natural language by using the PoS tagging method defined in RASP system [2] and develops the operational requirement of the cloud service requester by using the method defined in [13].

3.7 Cloud service searching agent

This agent will communicate the domain specifying agent and forwards the requirement of the cloud service consumer. The domain specifying agent will identify the domain of the particular requirement and send back the result. The cloud service searching agent will generate SPARQL query according to the requirement and search the CS ontology repository for the services and provide the list of cloud services as suggestions to the cloud service consumer. The query language that is used by this system is SPARQL [17].

4 Workflow of SACoSS

The workflow of SACoSS system can be divided into two divisions, Cloud Service Ontology Updation and Cloud Service Suggestion Generation. SACoSS system is capable of differentiate cloud services in terms of SaaS and IaaS only, where the constraints for defining a cloud service as PaaS service are not considered during the development of the system.

4.1 Cloud service ontology updation

The SLA processing agent has a sub agent called service crawler whose duty is to retrieves the SLA and WSDL document of the Cloud Service Providers from the web. The SLA processing agent process the documents through RASP system [2] and generates the environmental template of the cloud service by the natural language processing method defined in [13] where the technical and non functional attributes of cloud services are predefined and extracted, for this purpose around 320 attributes type have been identified and used. Then the service template is passed through Semantic Annotating agent which adds annotation to the words of the document by means of [12]. Then the template will be passed on to domain specifying agent. The domain specifying agent then identifies the domain of the particular service by matching the words of service template with predefined keywords of SaaS and IaaS services like software, storage, uptime of service etc. For matching purpose around 140 keywords has been identified to segregate a cloud service as IaaS or SaaS cloud service. The matching technique follows a simple rule - majority of keyword matches then the service domain is that.

Depend upon the service type the domain specifying agent then forwards the service template to the corresponding ontology updating agent (i.e. SaaS or IaaS). Both the ontology updating agents follows the updating process to update the cloud service ontology, where the cloud service ontology structure is developed based on OWL-S [22]. To reduce the complexity, without non functional attributes, the reduced cloud service ontology structure for SaaS services is given in figure 2. SaaS cloud service ontology contains classes for representing the software details like browser support, minimum hardware requirement, user credentials service type, usage limit etc.,
along with non functional attributes.

Fig. 2: SaaS ontology structure

The reduced cloud service ontology structure for IaaS services without non functional attributes is given in figure 3. IaaS cloud service ontology consists of classes for representing infrastructure details like software and hardware requirements, service type, user credentials, storage space limit, availability of the service, integrity constraint, encryption option, backup option etc., with non functional attributes. The ontology updating agents parse the cloud service templates and update the corresponding service attributes as properties of the ontology classes.

4.2 Cloud service suggestion generation

The SACoSS system takes query from the user in natural language. The system provides an option for adding annotation for the query; if the cloud service consumer is an expert user then the consumer can add annotation for the query. For example if consumer is in need of 5GB of storage space then consumer can provide annotation for the word storage like memory, raw disk etc. The query processing agent follows the same mechanism used by SLA processing agent and develops the query structure. Then the query is sent to the domain specifying agent which finds the domain of the query and forwards the query with appropriate domain to cloud service searching agent.

The cloud service searching agent generates the SPARQL query according to the query structure and submit the query to CS-ontology repository which in turn returns the appropriate results. The algorithm used to generate query and to retrieve result is given below

(1) For each request  
   a. Develop the object restrictions  
   b. Develop the restriction for datatype properties

(2) Develop the query  
   a. Input - set of objects with value attributes  
   b. Create select query for domain class

(1) If there is a condition
(2) Create Filter sub query for value attributes
(3) If there is a combination for service conditions
(4) Create Union sub query for value attributes
(3) Find appropriate ontology type depending on domain class of query
(4) Submit query and retrieve result
(5) Generate list of service in table format from result

The cloud service searching agent produces the services list in table format to the cloud service consumer.

5 Performance Evaluation of SACoSS

SACoSS system has been implemented using Jena semantic web framework [10] and SPARQL for querying the cloud service ontology repository through ARQ. The cloud service ontology repository is a MySQL database which is accessed through Jena for the storage of service ontology. Apache Tomcat was used to host the CS-ontology repository. The agents are given access through a JAVA API which has the functionality of querying. The list of services used by SACoSS is derived from [7] which provide details about cloud service providers under different categories. For SaaS service ontology, 15 domains have been identified and 178 services are classified according to their domains. For IaaS service ontology, 6 domains have been identified and 94 services are stored accordingly. To evaluate the performance of SACoSS, it has been tested in a Pentium Xeon, RedHat Linux server system and uses Semantic Web Service Matchmaker Evaluation Environment[20] for comparing the results with [16, 1] and OWLS-M4 [14]. The Semantic Web Service Matchmaker Evaluation Environment evaluates semantic service selection systems over given test collections in terms of standard retrieval performance evaluation measures and provides output through chart, graph, XML. The measures used to evaluate the SACoSS performance are query response time, precision recall and fallout. Figure 4 depicts precision recall curves comparison and figure 5 depicts fallout recall curves comparison. For comparing query response time, 20 queries have been given as input to all the four systems. The complexity of queries increases
for each one, the number of words and requirements for a service increases from query to query. Figure 6 depicts the comparison of query response time of all the four methods. The results reveals that SACoSS system outperforms other three methods in most of the cases and performs in par with them in few cases.

Fig. 4: Precision-recall curve

Fig. 5: Fallout-recall curve
6 Conclusion and Future Work

For a naive user of cloud computing, to select an appropriate cloud service is a difficult task as it involves understanding of SLA documents and WSDL documents which provides technical and non functional aspects of a cloud service. Hence to overcome this issue we presented a system called SACoSS for cloud service suggestion according to the requirements of the cloud service consumer by using cloud service ontology. SACoSS has been developed using various semantic technologies. This system comprises seven semantic agents and a cloud service ontology repository. The workflow of SACoSS can be divided in to two sections, cloud service ontology updation and cloud service suggestion generation. SACoSS system is capable of suggesting SaaS kind cloud service and IaaS kind cloud services. The performance evaluation of the SACoSS system reveals that it performs well than other similar kind of methods that has been proposed earlier.

![Query response time](image)

**Fig. 6**: Query response time

**Fig. 7**: Abbreviations

- **OWL-M4** – Hybrid Semantic Web Service Matchmaker
- **WSCO** – Web Services Selection Based on Context Ontology and Quality of Services
- **SASD** – Semantics Based Automated Service Discovery
- **SACoSS** – Semantic Agent based Cloud Service Suggestion system

In the view of extending SACoSS system, an ontology structure for PaaS kind of services can be developed and incorporated with it. SACoSS system does not provide any ranking to the services and a new ranking methodology for services can be developed and incorporated with SACoSS. While selecting a service, taking user reviews about that service in to account as a constraint could be a good practice. Hence developing and incorporating user reviews with each service in service ontology could become a good extension for SACoSS.
References


