Institute of Advanced Engineering and Science

International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.3, No.6, December2014, pp. 330~337
ISSN: 2089-3337

COBE Framework: Cloud Ontology Blackboard Environment for Enhancing Discovery Behavior

Ahmed Ghoneim[1, 3] & Amr Tolba[2, 3*]

[1]College of Computers and Information Sciences, King Saud University, Saudi Arabia,
[2]Riyadh Community College, King Saud University, Saudi Arabia
[3]Faculty of Science, Menoufia University, Egypt
{ghoneim, atolba}@ksu.edu.sa
*Corresponding Author

Article Info

Article history:

Received Oct 12th, 2014
Revised Oct 20th, 2014
Accepted Oct 29th, 2014

Keyword:

Ontology
cloud computing
agent systems
AI reasoning
design style

ABSTRACT

The new relatively concept of cloud computing & its associated methodologies has many advantages in the world of today. Such advantages range between providing solutions for integration of the miscellaneous systems & presenting as well guarantees for distribution of searching means & integration of software tools which are used by consumers & different providers. In this paper, we have constructed an ontology-based cloud framework with a view to identifying its external agent's interoperability. The proposed framework has been designed using the blackboard design style. This framework is composed of mainly two components: controller and cloud ontology blackboard environment. The function of the controller is to interact with consumers after receipt of the subject request where it spontaneously uses the ontology base to distribute it & constitute the required related responses whereas the function of the second framework component is to interact with different cloud providers and systems, using the meta-ontology framework to restructure data via using AI reasoning tools and map them to its corresponding redistributed request. Finally, E-tourism case study can be applicable will be explored.

Corresponding Author:
Amr Tolba
Riyadh Community College, King Saud University,
Faculty of Science, Menoufia University,
Email: atolba@ksu.edu.sa

1. INTRODUCTION

Cloud computing has been formulated as a general term to describe a class of complex computing services on demand, which was introduced by several commercial providers, such as Amazon, Google, and Microsoft. It denotes a model that looks at the infrastructure of computing as a "cloud", through which companies and individuals can access applications from anywhere in the world based on their demand. The main principle behind this computing model is to provide computing, storage, and software "as a service" [8].

The most common service models introduced in the cloud computing environment are IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). In IaaS model, the service providers outsource the equipment used to support operations, including hardware, storage, networking components and servers over the internet. In PaaS model, the service providers allow the cloud consumers to use operating systems, and virtualized servers together with its associated services for running applications over the internet. Finally, In SaaS model, the service providers allow cloud users to remote access of applications or software as a web-based service to avoid the cost of buying software [10, 11].

Ontology aspect is used to constitute basic- and processed- information which describe the data semantics based on the consumers request. It has a set of concepts that deal with information retrievals to process the consumers and providers requests [4, 7]. Most of the existing cloud service discovery frameworks are using the ontology reasoning for matchmaking the suitable requests and responses for both consumers and providers [4, 9].

In this paper, we introduce an ontology-based cloud framework for discovery their external agent's interoperability. The proposed framework is designed based on blackboard design style. This framework is composed of mainly two components: controller and cloud ontology blackboard environment. Both components are using ontology for discovering the best suit set of responses regarding the consumer requests.

The rest of this paper is organized as follows: In Section 2, the proposed framework is described; its structural and behavioral is illustrated in details. Section 3 illustrates the related works. Case study is described in section 4. Finally, In Section 5, conclusions and future work are discussed

2. PROPOSED FRAMEWORK

In this section, we have been introducing both the structure and behavior of our ontology-based cloud framework for discovery their external agent's interoperability. The proposed framework is designed based on blackboard design style. This framework is composed of mainly two components: controller and cloud ontology blackboard environment as shown in Figure 1.

The role of controller is to directly interact with consumers. The controller functionality starts, after receiving the consumer request, and then it uses its built-in controller-ontology for decomposing the request into set of mini-requests (i.e. distributing the request to its primary basic components). After that, send this set of requests to the cloud ontology blackboard environment component and waiting for responses. Finally, after receiving the responses from the (COBE) component, it starts his final process by constitute these set of responses into one consistence response and send it back to its consumer.

The (COBE) component is responsible for interacting to all possible clouds, different agent systems, and controller. It uses meta-ontology for restructuring data, using AI reasoning methods, and mapping data to its equivalent redistributed request. The behavior of (COBE) consists of the following sequence: first, receive the distributed set of request from the controller, then using the meta-ontology for restructuring cloud and agent data and runtime updating this data for any changes. After that, it uses AI reasoning methods such as forward or backward reasoning to extract the acceptable responses from the data streaming flow between COBE and its external agents. Next, COBE starts its processes in this stage by mapping the responses data to its equivalent request. Finally, COBE sends the response sets and its mapping parameters to the controller. The implemented behavior of COBE is illustrated as pseudo code in table 1.

The COBE starts its functionalities by checking the request type; if the request type is simple this means the COBE neglecting the distribution option. In case the request type is composite this implies the distribute function of COBE will be fire, and based on the result of distribution, the COBE starts the second function by mapping each sub request to its partner.

The main component of the proposed framework and its interaction are shown in Figure 2, which represents the structure part of the COBE. The dynamic behavior of the COBE is illustrated using Figure 3, which focuses on business processes and dataflow of the system Moreover each component of the core uses the business rules to handle the flow or apply the reasoning methods.

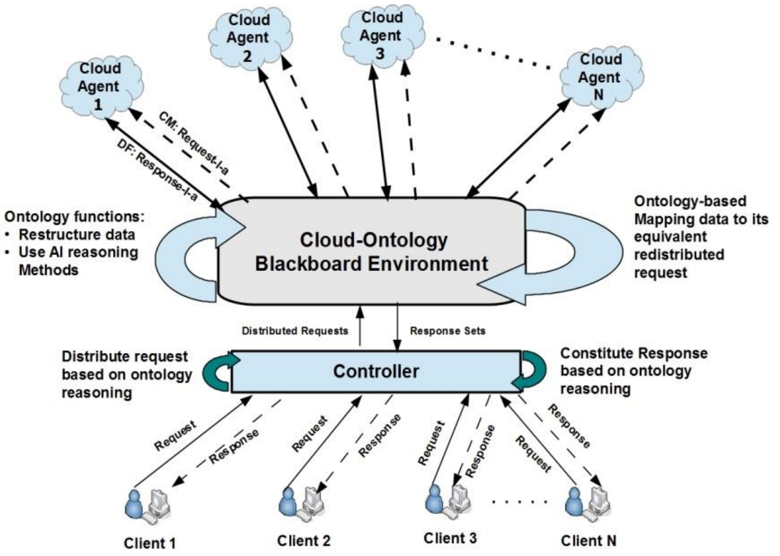


Figure 1 – COBE Framework

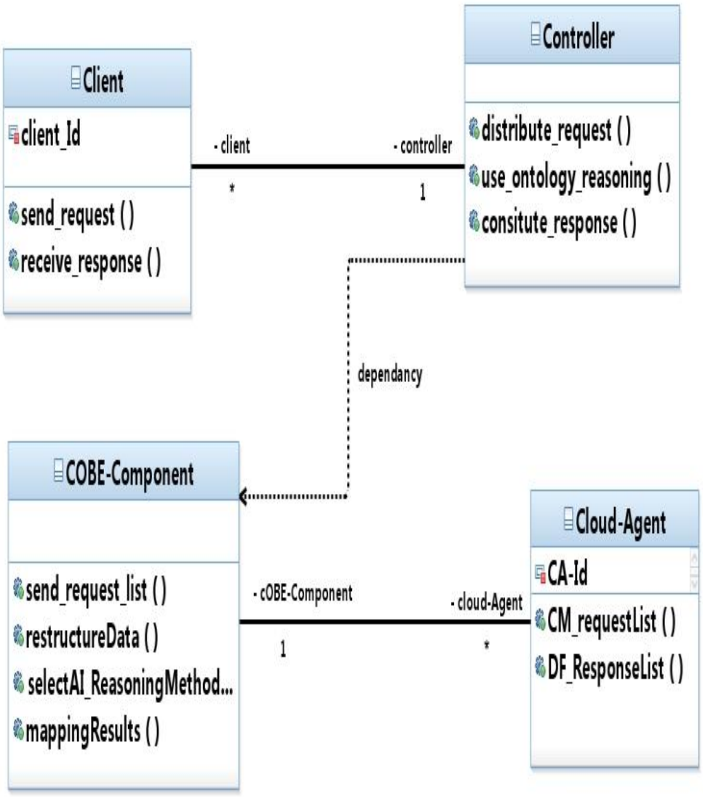


Figure 2: Class diagram of the proposed framework

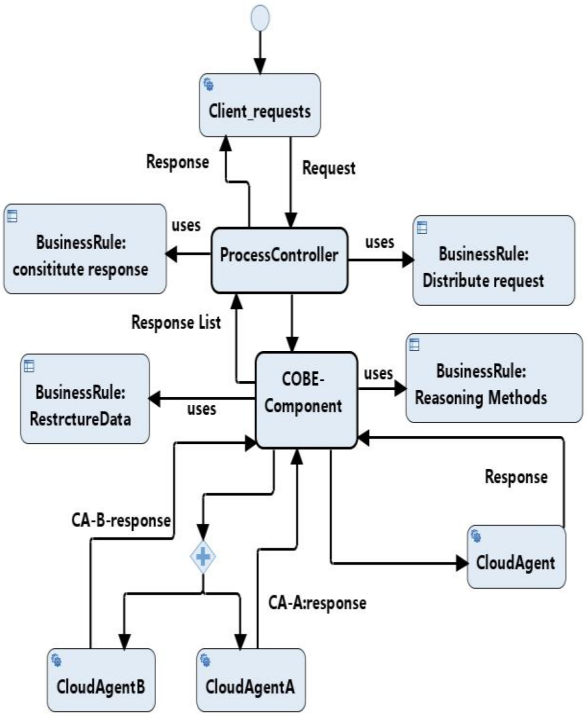


Figure 3: Business processes that represent the internal functionality of COBE framework

```
Inputs:      Consumer requests, set of requests
Out puts:   Equivalent Responses

BEGIN- Main
For All Consumers
BEGIN
COBE_Controller.recive(requests)
END FOR
For each request
COBE_Controller.decompose
         (Ontology,(IN:requests,OUT:set of mini-reqests))
END FOR

COBE_Controller.send (set of requests)
COBE.receive (set of requests)
COBE.restructure (Agent Data, runtime)
COBE.uses((reasoning methods), set of requests)
COBE.response-mapping(set of responses, set of requests)
COBE.send(mapping output, Controller)

For each set of response
COBE_Controller.constitute-response(mappingoutputs)
END FOR

For each set of constitute response
COBE_send (response, equivalent-consumer )
END FOR

END-Main
```

Table 1: Pseudo code implemented behavior of COBE

3. RELATED WORK

In the following, we will illustrate some related work for applying ontology concept in a cloud computing environment. Lamia Youseff et al. [1] proposed the use of ontology inside the cloud. They try to establish a knowledge domain in the field of cloud computing. They present cloud ontology as a stack of layers discussed each layer in the strength point of view and neglect the integration and communication issue between layers.

N.M.Karia et al. [2] have focused on introducing an ontological framework that provides a structure of different cloud environments and, describing its components. This work doesn't address the heterogeneity of terms used by vendors, number of standards, and a structure of the different cloud environments. Teodor-Forin Forties et al. [3] attempt to achieve ontology concept in the cloud services. They introduce mOSAIC as a middleware to facilitate the communication with various cloud providers. They suggest the use of ontology to facilitate the selection, execution and discovery of resources and services. They introduce a comprehensive model of ontology cloud the can work only for public clouds.

Taekgyeong Han and Kwang Mong Sim. [4] present a Cloud Service Discovery System (CSDS) based on ontology. They use ontology to enhance performance of the system also, the use of software agent to consult ontology when selecting information retrieved from a public cloud. Miranda Zhang et al. [5] concentrated on using ontology for Cloud Based IaaS services. They implement ontology in the CloudRecommender system with focusing on IaaS services and neglect PaaS and SaaS configurations in a public cloud. On the other hand, lot of work done to utilize a software agent in cloud computing environment to enhance the performance of cloud services. Toshihiro Uchibayashi et al. [6] introduced a framework of an agent-based support system for discovering services in a public cloud. They use agents implemented by JADE only to measure network usage against user requirements information.

Based on the above related works, we can realize that most of the above models and frameworks are using the AI reasoning methods and ontology without providing an approach on how to distribute the request into request list, restructure the cloud data at runtime and mapping the parts of the request list into its equivalent partner. Our proposed COBE framework addresses the above challenges.

4. CASE STUDY: E-TOURISM

In this section, we will approve the applicability of our COBE framework by illustrating how the different systems and agents are communicating together through the blackboard style. Moreover, how to distribute the visitor request to asset of requests and mapping these set into its equivalent agents. In figure 4, the E-tourism data inputs and its processed output is designed using component based style. The required services and the provided services of the system are identified.

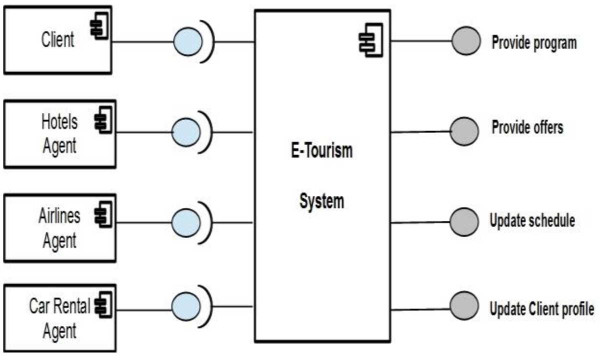


Figure 4: E-tourism system with required and provided data

The data center interoperability between the E-tourism systems is shown in figure 5, which represent the dataflow and the control flow from COBE main component into all interacted agents using blackboard style. COBE of E-tourism has the ability to detect and identify the set of agents based on the number of distributed requests. Moreover, COBE also identify the type of communication with these agents and play the role of controller to those agents. The number of agents depends on the number of sub-requests. The COBE sometimes find that one of the identified agents which is not connected to its functions not responding. In this case the COBE has a wide cloud search function to search for alternative agents with the same feature and compare between them and to select the best one.

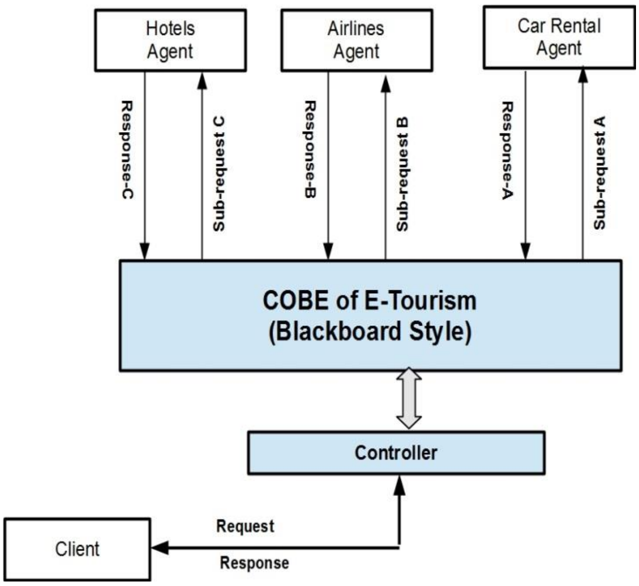
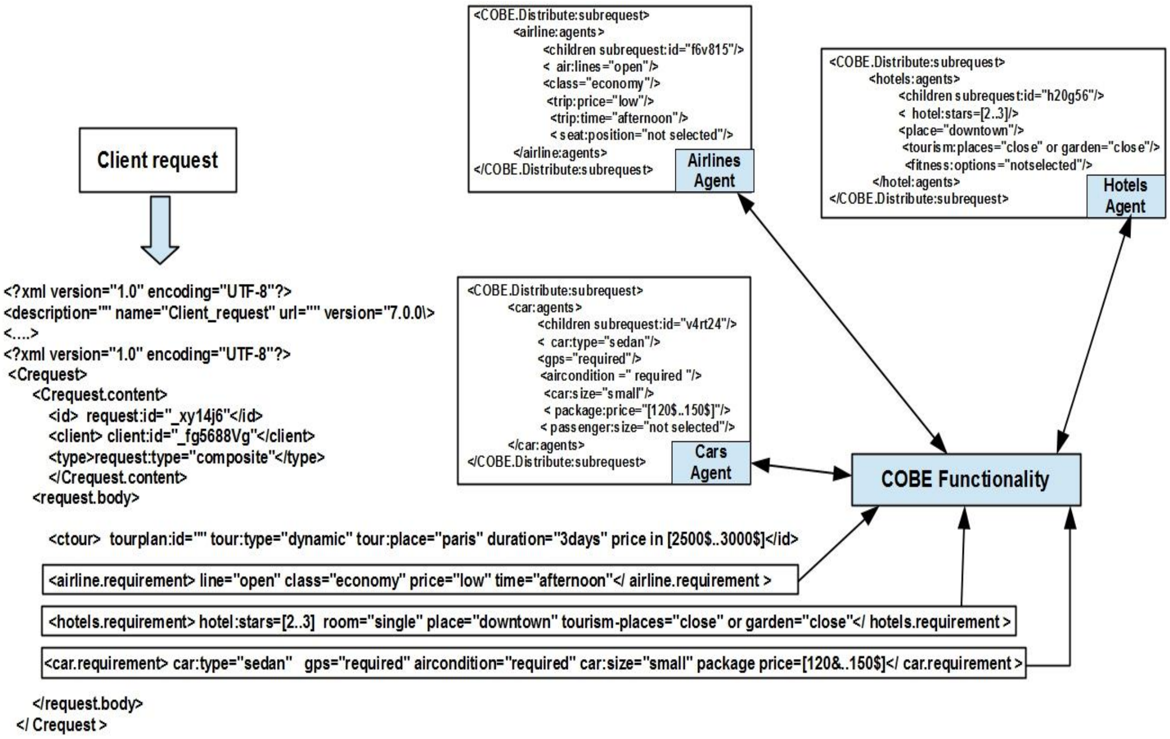


Figure 5: COBE Role on E-tourism

In Figure 6, we provide a sample of visitor request as XML message and the role of COBE to distribute it into set of sub-requests and its equivalent agents (Hotels-agent, Airlines-Agent, Car Rental - Company Agent). The COBE uses AI forward reasoning techniques to distribute the request to a set of consistent sub-requests as shown in samples at figure 6. After that, it waits the responses from all connected agents. Next, it uses the validation and verification techniques to check that each response is valid to its sub-request. Then, the COBE uses AI backward reasoning techniques for constituting the set of responses into one response including some features or plan options. Finally, The COBE provides this response to the identified client.



```
<COBE.Distribute:subrequest>
        <airline:agents>
                <children subrequest:id="f6v815"/>
                <  air:lines="open"/>
                <class="economy"/>
                 <trip:price="low"/>
                  <trip:time="afternoon"/>
                  < seat:position="not selected"/>
        </airline:agents>
</COBE.Distribute:subrequest>
```

```
<COBE.Distribute:subrequest>
        <hotels:agents>
                <children subrequest:id="h20g56"/>
                <  hotel:stars=[2..3]/>
                <place="downtown"/>
                 <tourism:places="close" or garden="close"/>
                  <fitness:options="notselected"/>
         </hotel:agents>
</COBE.Distribute:subrequest>
```

```
<COBE.Distribute:subrequest>
        <car:agents>
                <children subrequest:id="v4rt24"/>
                <  car:type="sedan"/>
                <gps="required"/>
                 <aircondition =" required "/>
                  <car:size="small"/>
                  < package:price="[120$..150$]"/>
                 < passenger:size="not selected"/>
        </car:agents>
</COBE.Distribute:subrequest>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<description="" name="Client_request" url="" version="7.0.0\>
<....>
<?xml version="1.0" encoding="UTF-8"?>
 <Crequest>
     <Crequest.content>
        <id>  request:id="_xy14j6"</id>
        <client> client:id="_fg5688Vg"</client>
        <type>request:type="composite"</type>
        </Crequest.content>
     <request.body>

        <ctour>  tourplan:id="" tour:type="dynamic" tour:place="paris" duration="3days" price in [2500$..3000$]</id>

      <airline.requirement> line="open" class="economy" price="low" time="afternoon"</ airline.requirement >

      <hotels.requirement> hotel:stars=[2..3]  room="single" place="downtown" tourism-places="close" or garden="close"</ hotels.requirement >

      <car.requirement> car:type="sedan"   gps="required" aircondition="required" car:size="small" package price=[120&..150$]</ car.requirement >

     </request.body>
  </ Crequest >
```

Figure 6: Interactivity between COBE of E-tourism Agent using XML messages

5. CONCLUSION AND FUTURE WORK

In this paper, we presented the COBE framework which provides the ability to manage the client request by distributing it to mini-requests and use the ontology with reasoning methods for identifying the responses for all the request sets by mapping it to its equivalent cloud agents. The propose framework is composed of two core components: Controller and COBE, both components are responsible for driving the framework functionalities. The structure and the business processes simulation of the proposed framework are designed. Moreover, the implemented pseudo code as algorithmic environment of the COBE framework is illustrated. In future work we are planning to use the quality of service (QoS) criteria to measure the response-time, availability and throughput. Moreover, we will integrate with CloudSim tool as a simulation environment for the COBE framework.

REFERENCES

[1] Youseff, Lamia, Maria Butrico, and Dilma Da Silva. "Toward a unified ontology of cloud computing." Grid Computing Environments Workshop, 2008. GCE'08. IEEE, 2008.

[2] Karie, Nickson M., and Hein S. Venter. "An Ontological Framework for a Cloud Forensic Environment." (2013).

[3] Fortis, T-F., Victor Ion Munteanu, and Viorel Negru. "Towards an ontology for cloud services." 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), IEEE, 2012.

[4] Han, Taekgyeong, and Kwang Mong Sim. "An ontology-enhanced cloud service discovery system." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2010.

[5] Zhang, Miranda, et al. "An Ontology based System for Cloud Infrastructure Services Discovery." arXiv preprint arXiv:1212.0156 (2012).

[6] Uchibayashi, Toshihiro, Bernady O. Apduhan, and Norio Shiratori. "A framework of an agent-based support system for IaaS service discovery." 13th International Conference on Computational Science and Its Applications (ICCSA), IEEE, 2013.

[7] Dieter, McGuiness, D.L. , Schulten, E. , Wee Keong Ng , Ge Peng Lim , and Guanghao Yan. "Ontologies and electronic commerce." Intelligent Systems, IEEE 16.1, 2001.

[8] Amr Tolba, and Ahmed Ghoneim “IABCF Smarter: An Intelligent Agent-Based Cloud Framework”, IJCST Vol. 3, Issue 4, Oct - Dec 2012.

[9] Liu, L., Yao, X., Qin, L., & Zhang, M. “Ontology-based service matching in cloud computing”. IEEE International Conference on In Fuzzy Systems (FUZZ-IEEE), (pp. 2544-2550). IEEE. 2014.

[10] Rafique, Ansar, et al. "Towards portability and interoperability support in middleware for hybrid clouds." IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE. 2014.

[11] Mosbah, Mohamed M., Hany Sol, and Mohamad Abou El-Nasr. "Cloud Computing Framework for Solving Egyptian Higher Education." Fourth International IEEE Conference on Advances in Computing and Communications (ICACC), IEEE, 2014.

BIBLIOGRAPHY OF AUTHORS (10 PT)

	Dr. Ahmed Ghoneim received his M.Sc. degree in software modeling from University of Menoufia, Egypt, and the Ph.D. degree from the University of Magdeburg (Germany) in the area of software engineering, in 1999 and 2007 respectively. He is currently an assistant professor at the department of software engineering, king Saud University. His research activities address software evolution; service oriented engineering, software development methodologies, scripting languages.
	Dr. Amr Tolba received his M.Sc. and Ph.D. degrees in computer science, Menoufia University, Egypt, in the area of databases and Multi-Agent systems, in 2002 and 2006 respectively. He joined the king Saud University (KSU) in August 2006 as an assistant professor at the department of computer science. His research interests include databases, multi-agent systems, system analysis and design, and cloud computing