**Research Article**

# Ontology-based Cloud Services Representation

[1]Abdullah Ali, [1]Siti Mariyam Shamsuddin and [2]Fathy E. Eassa
[1]Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia,
81310 Johor, Malaysia
[2]Department of Computer Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

**Abstract:** The advancement of cloud computing has enabled service providers to provide diversity of cloud services to users with different attributes at a range of costs. Finding the suitable service from the increasing numbers of cloud services that satisfy the user requirements such as performance, cost and security has become a big challenge. The variety on services description none uniformed naming conventions and the heterogeneous types and features of cloud services led to make the cloud service discovery a hard problem. Therefore, an intelligent service discovery system is necessary for searching and retrieving appropriate services accurately and quickly. Many studies have been conducted to discover the cloud services using different techniques, such as ontology model and agents technology. The existing ontology for cloud services does not cover the cloud concepts and it is intended to be used for specific tasks only. This study represents the cloud concepts in a comprehensive way that can be used for cloud services discovery or cloud computing management.

**Keywords:** Cloud computing, cloud computing management, cloud services, cloud services discovery, cloud services representation, ontology

## INTRODUCTION

The Internet has become one of the most important communication media in the world due to its usage in various life areas such as e-business, education, health, e-governments, social networking and other services. The rapid development of information technology in the recent years motivates many organizations and enterprises to use searching methods to keep operational cost, achieve scalability, good performance and high efficiency in resource utilization (Soundararajan and Govil, 2010). Distributed systems, parallel computing, grid computing, virtualization and other technologies have evolved over the years. The inflexibility, high cost and deficiency of scalability of these technologies are ineffective for business requirements. Recently, cloud computing has emerged to meet business requirements by trying to complement these technologies and add new features to resources and application provisioning.

Cloud computing is a model that allows users to use the hosted services (hardware and software) that are available over the Internet. Recently, the majority of popular sites are hosted on a cloud, including social networking, email, document sharing and online gaming (Shimba, 2010). There are two players in cloud computing, cloud providers and cloud consumers. The cloud providers keep enormous computing resources (services) in their large datacenters and rent these services to consumers on a pay-per-use basis (Lee, 2012). The development of Service-Oriented Architecture (SOA) and cloud computing, have arisen a large number of service providers that offer a diversity of services for users with different performance attributes and costs which are used widely in every day, including the virtual hosting services, storage services and Web application services (Zhao *et al*., 2012; Yongbo and Ruili, 2011). The cloud customers have faced a problem of how to find the best cloud service from the growing numbers of cloud providers that satisfy the QoS requirements such as performance, cost and security (Garg *et al*., 2013).

Finding the convenient service that satisfies the user requirements based on both functional and non-functional requirements have become a big challenge (Wu *et al*., 2012). In general, the user can use the search engine, such as Google to search for a suitable service. But, it is not effective and efficient enough specifically if the services are similar but with different attributes (Chang *et al*., 2012). The general search engine has been developed for searching on published documents and it is based on keyword search whereas the data on the cloud computing are considered as unpublished documents. So, it is necessary to search for these important cloud data that it is hard for general search engine, as a Google, to search it (Miyano and

**Corresponding Author:** Abdullah Ali, Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor, Malaysia

Uehara, 2012). Unfortunately, the users cannot quickly search all kinds of current cloud services due to the absence of service discovery mechanism, open protocol, or specific criteria (Zhang *et al*., 2013). The traditional Web Service Definition Language (WSDL) describes the service from technical viewpoint including features of service, interface operations, binding, etc., (Chinnici *et al*., 2007). These descriptions are not sufficient for describing the services that are delivered via the Internet from a business viewpoint and in which the cloud computing services are of business service type. Business service is concerned with the end to end delivery between the provider and customers. It depends on a particular period of time, cost and service level agreements which represent the important parameters for searching for the cloud services (Cardoso *et al*., 2010; Zeng *et al*., 2009).

Despite the considerable amount of research works on addressing the various challenges in cloud computing such as data processing and migration (Hausenblas *et al*., 2012; Menzel and Ranjan, 2012; Truong *et al*., 2012), knowledge management (Fehling *et al*., 2012), accessibility (White, 2011) and security and privacy (Ren *et al*., 2012; Pearson and Benameur, 2010), cloud services discovery still largely remains an untouched area (Wei and Blake, 2010).

The limitations of the traditional search tools and the problem of matching between user requirements and services advertised by cloud providers can be overcome through ontologies and semantic technologies (Lupiani-Ruiz *et al*., 2011). It is also useful for information retrieval (Valencia-García *et al*., 2008), information integration (Wang *et al*., 2006), service discovery (Castells *et al*., 2007), question answering (Valencia-García *et al*., 2011), recommendation (García-Crespo *et al*., 2011) and information management (Colomo-Palacios *et al*., 2010).

Existing cloud computing ontologies are mostly general and detailed ontologies of each cloud computing are still missing (Androcec *et al*., 2012). With the help of ontology, they can deal with various types of queries and allocate resources suitable for the type of services and description of jobs requested from cloud users. Cloud users request a job with detailed user requirements, such as deadline, budget, CPU size, type of operating system, storage size and QoS parameters, response time, reliability, availability and so on (Ma *et al*., 2011). The Cloud Services Ontology (CSO) describes data semantics of cloud services, which is critical in the sense that cloud services may not necessarily use identifying words (e.g., cloud, infrastructure, platform and software) in their names and descriptions (Noor *et al*., 2013).

This study represents the cloud concepts in a semantic meaning using ontology which can be used for cloud services discovery or cloud computing management. The developed ontology will improve a query searching mechanism for selecting the most relevant cloud services.

## LITERATURE REVIEW

In information systems, ontology is used to describe knowledge in a certain domain. Ontologies consist of hierarchical definitions of important concepts in a domain and descriptions of the properties of each concept. Usually suitable terminologies and the semantic properties are expressed in the form of the ontologies (Gruninger and Lee, 2002). Ontology in information systems used to represent the knowledge of particular domain in a hierarchical form. The significant concepts and the characteristics description of every concept are determined by this hierarchy. Ontologies have three major uses, which are: to accomplish interoperability, facilitate the communications among software systems and to assist the communication among humans (Jasper and Uschold, 1999; Maedche and Staab, 2001). Recently, the cloud computing services have been increasing rapidly. Therefore, the need for a taxonomy framework has become necessary to classify the cloud services based on their attributes such that they become easy for understanding and comparing them. There are common characteristics among cloud services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) and there are specific characteristics for each one. The common characteristics include license type, user group, payment mode, security measures etc. Some specific characteristics of IaaS are supported by the operating systems and/or virtualization technology. The PaaS services have specific attributes such as languages, environments and operating systems supports. The SaaS services have a specific characteristic customer/application domain (Hoefer and Karagiannis, 2010). Ontology defines a common vocabulary for information sharing in a domain. The definition of common and shared domain concepts in the ontologies enables machines to exchange semantics along with the syntax (Maedche and Staab, 2001). In a service discovery or resource management for cloud computing, the speedy and accurate services acquisition plays an important role in the management of the cloud computing resources. The decision that will be made based on the inaccurate knowledge or delay in choosing the services (resources) will affect the performance and may affect other factors too.

**Related works:** Many researchers have developed systems to allow users to find the suitable services that satisfy the desired needs. They have also developed many algorithms for resources management. Many of them have used the ontology model to represent the cloud services and to perform the process of matching between the available services and the desired needs (Kang and Sim, 2010; Sim, 2012; Chang *et al*., 2012). Ontology can be defined as a formal, "explicit specification of a shared conceptualization" (Studer *et al*., 1998; Kang and Sim, 2010). Ontology contains a set of concepts on the domain and the relationships

between these concepts. It can be applied into information retrieval to deal with user queries (Reshma and Saravana Balaji, 2012). Ontology (Shih *et al.*, 2009; Flahive *et al.*, 2006, 2009; ZadJabbari *et al.*, 2010) refers to representing and understanding of some domain concepts. It has been presented as a useful model for enabling system intelligence and improving the system ability to obtain system semantics.

The author's work (Rodríguez-García *et al.*, 2013) has dealt with two major issues:

- The semantic annotation of the features of cloud services
- The discovery of cloud services that meet users' needs

They built a semantic search engine that leverages the cloud service related annotation which is added as metadata to the cloud services description on the web contents in order to improve the precision and recall of the search results. Detailed cloud ontology is presented by Youseff, *et al.* (2008) which demonstrates the classification of cloud into five layers. These five layers of the cloud services are categorized as Hardware as a Service (Haas), Software kernel, Cloud Software Infrastructure comprised of Iaas, Data Storage as a Service (DaaS) and Communication as a Service (CaaS), Cloud Software Environment (PaaS) and Cloud Application (SaaS). This study covers only the delivery model of the cloud computing and discusses every layer's strengths, boundaries and reliance on preceding computing perceptions. The classification of cloud services and pricing models are proposed by Weinhardt *et al.* (2009). They Weinhardt *et al.* (2009) presented the three layers of ontology:

- Infrastructure
- Platform
- Application as a service

It can be used by the Cloud users and providers to map the present cloud services and can put pricing schemes. The work by Dastjerdi *et al.* (2010) provides an ontology-based discovery for QoS aware deployment of appliances on IaaS providers. An ontology-enhanced Cloud service discovery system is proposed by Han and Sim (2010). The system enables users to select Cloud providers based on the provided ontology. The authors (Tahamtan *et al.*, 2012) claimed that their ontology is much better in providing querying possibilities and is more comprehensive than the existing works where their work considers the three layers IaaS, PaaS and SaaS and it focuses on ontology based discovery of Cloud providers. Our work is a complement for the Tahamtan *et al.* (2012) that will be used for cloud services discovery and cloud resource management and it is more comprehensive.

An ontology-based resource management of Cloud providers is proposed by Ma *et al.* (2011). A Recommendation System (RS) is described by Han

*et al.* (2009) for cloud computing. This approach is most suitable for design time decisions as it is used statically to provide a ranking of available cloud providers. Smit *et al.* (2012) presented a methodology for the implementation of a service-oriented application which presents relevant metadata information describing cloud services via a uniform RESTful web service. The mOSAIC project developed by Moscato *et al.* (2011) defining a common ontology, aims at developing an open-source platform that enables applications to negotiate Cloud services as requested by users. The ontology techniques to find cloud services closer to cloud service consumers' requirements are exploited by proposing a Cloud Service Discovery System (CSDS) by Kang and Sim (2011). Particularly, Kang and Sim proposed a cloud ontology where agents are used to perform several reasoning methods such as similarity reasoning, equivalent reasoning and numerical reasoning. In addition, Noor *et al.* (2013) have used different cloud services ontology with different relations (is-a) and (is-not-a) relations to increase the accuracy of the discovery results. This study of Noor *et al.* (2013) is complementary to Kang and Sim (2011) for use in bigger environments (i.e., the world wide web) where they have developed a Cloud Service Crawler Engine, that is used to collect metadata related to cloud services through search engines.

This study is complementary for the existing works representing the cloud computing activities such as service delivery models, deployment models and other activities in a comprehensive ontology. It also covers the Quality of Servicr (QoS) and Service Level Agreement (SLA) requirements. This study can be used for different purposes including cloud services discovery and cloud resources management and it can be used during design time or at runtime decisions.

## METHODOLOGY

Different authors have used different methodologies for the complementation of their research. Grüninger and Fox (1995) have established different conditions for characterizing the completeness of the ontology. According to Noy and McGuinness (2001), there is no single correct way to model domain ontology, as there are always alternative ways to model it. Methodology used by Ahmed *et al.* (2007) focuses on the user domain but does not study the inter-relations between these concepts. Domain ontology for a product family is proposed by Kumara (2006) to identify artifacts. Many authors have used the methodology called the METHONTOLOGY (Fernández-López *et al.*, 1997) for the development of their ontology. This methodology (METHONTOLOGY) is based on the idea of software engineering which defines a set of tasks to be performed for developing a consistent and complete conceptual model. Moreover, METHONTOLOGY helps in building ontologies from scratch and it can be applied for the reuse of existing ontologies. So we have

also chosen this methodology (METHONTOLOGY) for this research due to the aforementioned reasons.

**Ontology construction:** Ontologies are being commonly used to bring semantics to the World-Wide Web (WWW). The WWW Consortium (W3C) developed the Resource Description Framework (RDF) (Brickley and Guha, 2000), a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The Defense Advanced Research Projects Agency (DARPA), in conjunction with the W3C, developed DARPA Agent Markup Language (DAML) by extending RDF with more expressive constructs aimed at facilitating agent interaction on the Web (Hendler and McGuinness, 2000). More recently, the W3CWeb Ontology Working Group developed Web Ontology Language (OWL) (Bechhofer *et al.*, 2004) based on description logic, maintaining as much compatibility as possible with the existing languages, including RDF and DAML.

The knowledge in ontologies is mainly formalized and five kinds of components are used: classes, relations, attributes, axioms and instances. The process of ontology construction consists of many phases. The first phase is the requirement analysis which specifies the concepts of ontology, attributes of concepts, relations between the concepts, axioms and the instances. A consistent conceptual model is determined in the design phase. In the development phase, the ontology formalization is performed using an appropriate ontology language which can assist the ontology to be updated in the phase of maintenance according to the target domain concepts (Zeshan and

Table 1: Some of cloud computing concepts

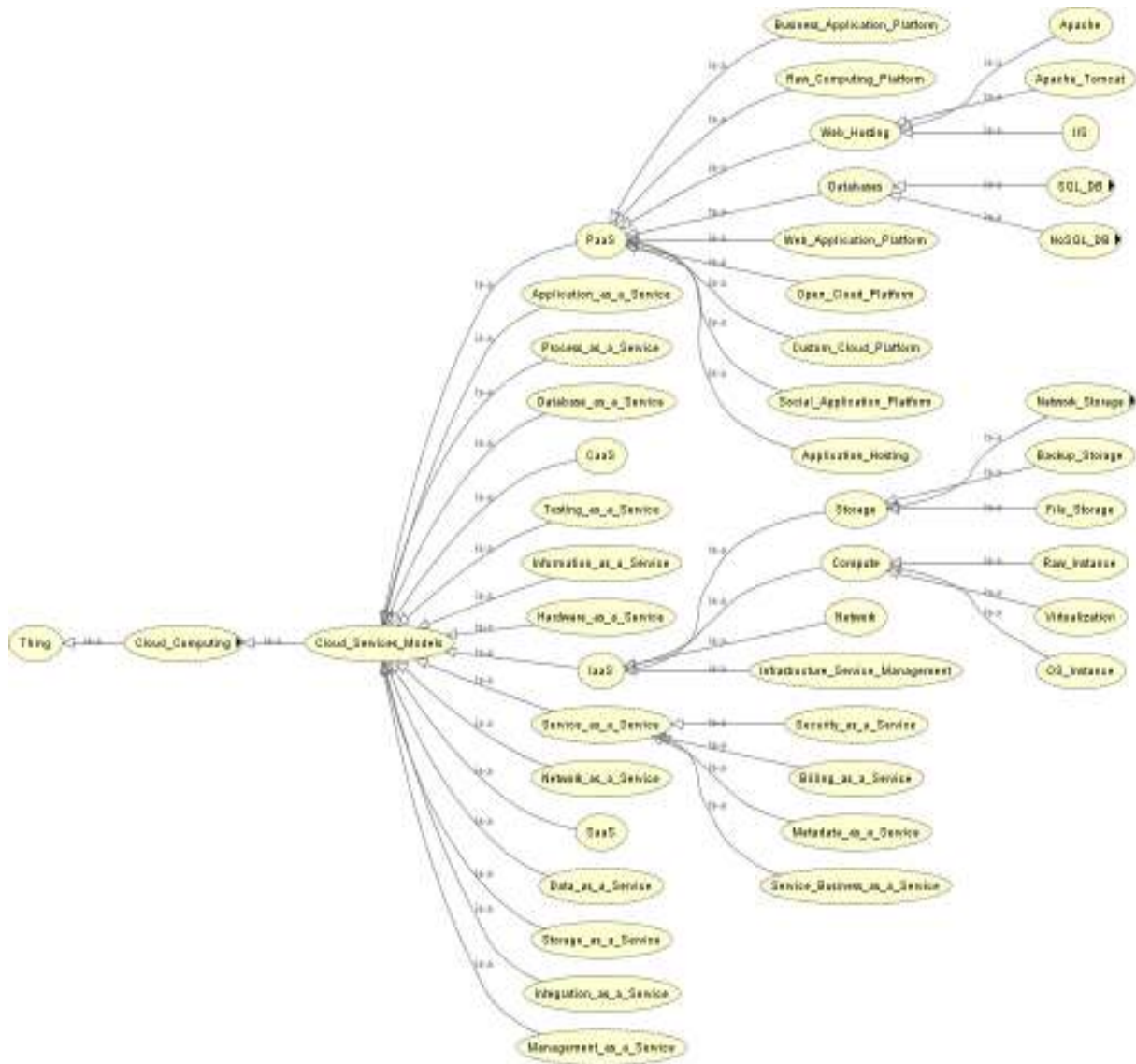| Cloud services concepts | | | |
|---|---|---|---|
| IaaS | Applicaction_hosting | OS_platform | Compute |
| DaaS | Deployment_model | Application | Software |
| PaaS | Physical_resources | Storage | Payment |
| SaaS | Cloud_computing_resources | Protocol | Security |
| CaaS | Delivery_model | Software | License |
| HaaS | Virtualized_resources | Games | Windows |
| SLA | Language_supported | Mail | Linux |
| QoS | Social_networking | Dot_net | Java |
| CPU | Web_hosting | Pythont | PHP |
| Ajax | Google_apps | ASP | S3 |
| EC2 | Google_docs | Public | Private |
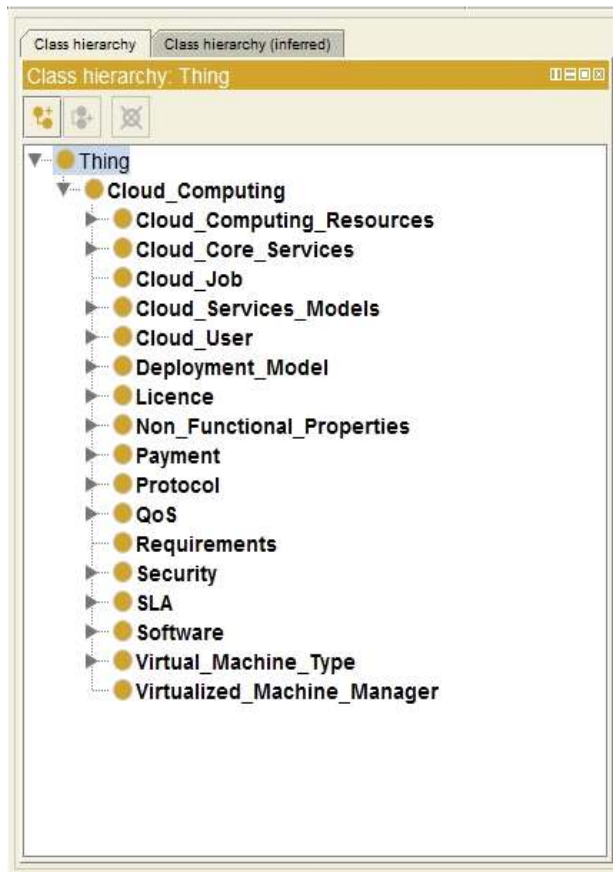
Fig. 1: Cloud computing ontology

Mohamad, 2012). The steps of cloud services ontology construction are achieved as follows.

**Entity extraction:** Extracting the entities is the process of discovering the concepts, the characteristics of these concepts and relationships among them. These concepts are collected from different relevant published papers (Hoefer and Karagiannis, 2010; Youseff *et al*., 2008; Tahamtan *et al*., 2012; Moscato *et al*., 2011; Ma *et al*., 2011; Wei and Blake, 2012; Zhang *et al*., 2012; Smit *et al*., 2012; John, 2009; Weinhardt *et al*., 2009; Han and Sim, 2011; Han and Sim, 2011; Chang *et al*., 2012). Table 1 provides some of the concepts related to cloud computing services.

**Taxonomy formation:** The concepts are arranged by the taxonomy in a hierarchical way in order to provide

the ontology with a structure to be understood by a human and to integrate it with other ontologies. The property "is-a" is used to define the relation among different domain concepts. The taxonomy and Hieracrhy of cloud services ontology is presented in Fig. 1.

**Concepts relationships:** Relationship among the domain concepts is determined by the properties and attributes that distinguish the domain classes. There are two types of properties: object property and data property. The relationship between two individuals is defined by object property, whereas data property is used to define the relationship between individual and XML Schema data type or RDF literal. The object properties and data properties for the cloud computing ontology are shown in Fig. 2.
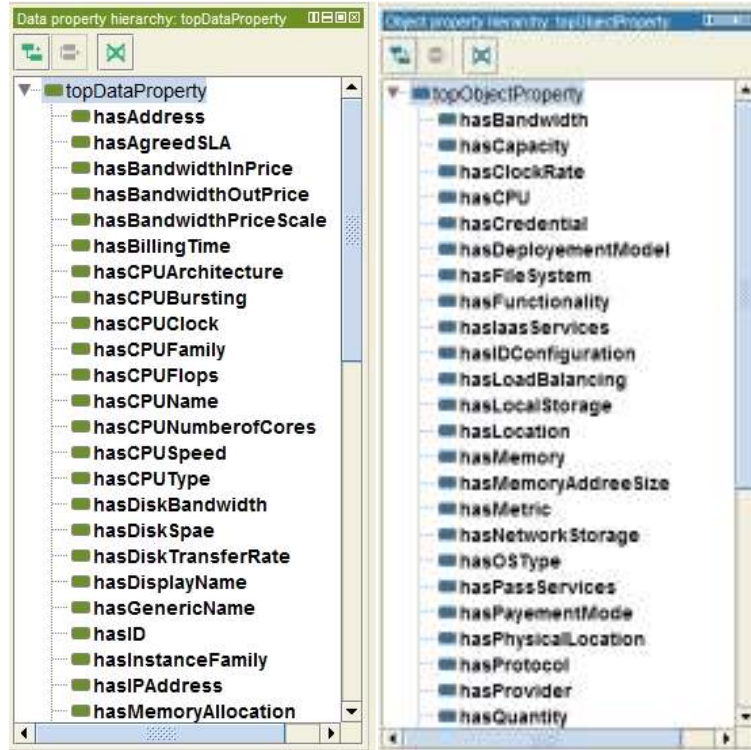
Fig. 2: Data and object properties

Table 2: Logical axioms for the cloud ontology

| Concept name | Axiom description | Logical expression |
|---|---|---|
| Physical_resources | A hardware-based device, such as a CPU, network or storage servers | Physical_resources ⊑ HardwarComponent ⊓ ∃usededFor.Computing ⊓ ∃ usededFor.AccessingNetwork ⊓∃ usededFor.Storage |
| CPU | A Central Processing Unit (CPU) is hardware that anything needs to be computed is sent to it | CPU ⊑ ProcessingUnit ⊓ ∃uses.Resoruces ⊓ ∃belongsTo.Physical_Resources ⊓usedFor.DataProcessing |
| Cloud_service | Software designed to perform a task. | Cloud_Service ⊑ hasFunctionality ⊓ ∃offeredBy.Cloud_Computing ⊓ ∃requestedBy.Cloud_User |
| Operating system | A collection of software which manages physical resources, mobile and embedded devices and provides common services for programs | OperatingSystem ⊑ PhysicalResourcesManagedSoftware ⊓ ∃usedBy.PhysicalResources ⊓ ∃hasCategory.GeneralOS ⊓ ∃hasCategory.MobileOS ⊓ ∃hasCategory.RealTimeOS ⊓ ∃hasCategory.EmbeddedOS |
| General purpose OS | An operating system that used by physical resources | GeneralPurposeOperatingSystem ⊑ OperatingSystem ⊓ ∃usedBy. Physical _Resources |
| Real time operating system | An operating system used by real-time devices. | RealTime_OS⊑ OperatingSystem ⊓ ∃usedBy. RealTimeDevice |
| Mobile operating system | An operating system used by mobile devices. | MobileOS⊑ OperatingSystem ⊓∃usedBy. MobileDevice |
| Cloud services model | Services provided by the cloud computing providers for the users via the internet. They include different models such as SaaS, PaaS, IaaS, DaaS,…etc | CloudServicesModels⊑ Service ⊓ ∃providedBy.CloudProviders ⊓ ∃accessedVia.Internet ⊓ ∃usededBy.CloudUsers⊓ (∃hasServiceType.SaaS ⊓ ∃hasServiceType.PaaS ⊓hasServiceType.IaaS ⊓ ∃hasServiceType.DaaS ⊓hasServiceType.CaaS) ⊓ ∃provededBy.CloudProviders⊓ ∃hasPayement.PayementMode ⊓∃hasCost.Price |
| Infrastructure as a Service (SaaS) | A provision model which is provided and supported by cloud providers including computing, storage and network | IaaS ⊑ CloudServicesMpdels ⊓ ∃providedBy.CloudProviders ⊓ ∃supportedBy. CloudProviders ⊓∃include.Compute⊓ ∃include.Storage ⊓ ∃include.Network ⊓ ∃hasCPU.CPU ⊓ ∃hasMemory.memory ⊓ ∃hasStorage.Storage ⊓ ∃usedBy.Cloudusers⊓ ∃hasNetwork.network |
| Software as a Service (SaaS) | A software model that the applications are hosted by the cloud providers to be available for the cloud users through the Internet | SaaS ⊑ Software ⊓ ∃hostedBy.CloudProviders ⊓ ∃usedBy. CloudUsers |
| Platform as a Service (SaaS) | Software and product development tools hosted in the provider's infrastructure and used by the developers through the Internet to create applications | PaaS ⊑ CloudServicesMpdels ⊓ ∃hostededBy.CloudProviders ⊓ ∃hostedIn. CloudProviderInfrastructure ⊓ ∃include.Software ⊓ ∃include.DevelopmentTools ⊓ ∃usedeBy.developer ⊓∃hasPurposeTo.createApplications |

**Axioms:** Axioms provide a proper way to add logical expressions to ontology. Such logical expressions can be used to refine the concept and relationships in the ontology. Axioms are used to design an explicit way of expression that is always true. Axioms can be used for defining the meaning of several components of the ontology, defining complex relationships and verifying the correctness of the information or obtaining new information. Table 2 presents the axioms for the cloud computing ontology.

**Cloud services individuals:** Individuals or instances represent the objects in the domain. In this study the individuals represent the cloud services provided by cloud providers. Figure 3 shows some individuals of the cloud ontology.

**Consistency checking:** Checking the consistency of ontology is very necessary as it detects the duplicating individuals which may be reducing the usefulness of the ontology. In order to perform this checking, the Protégé
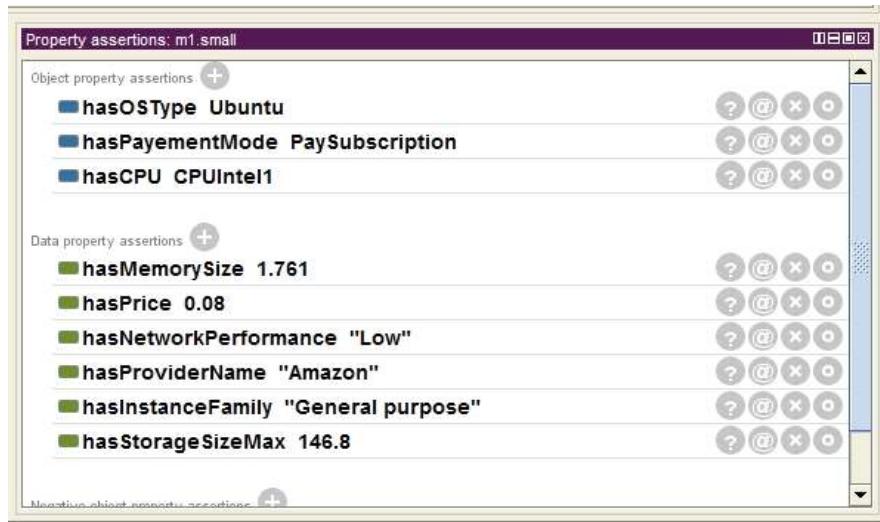


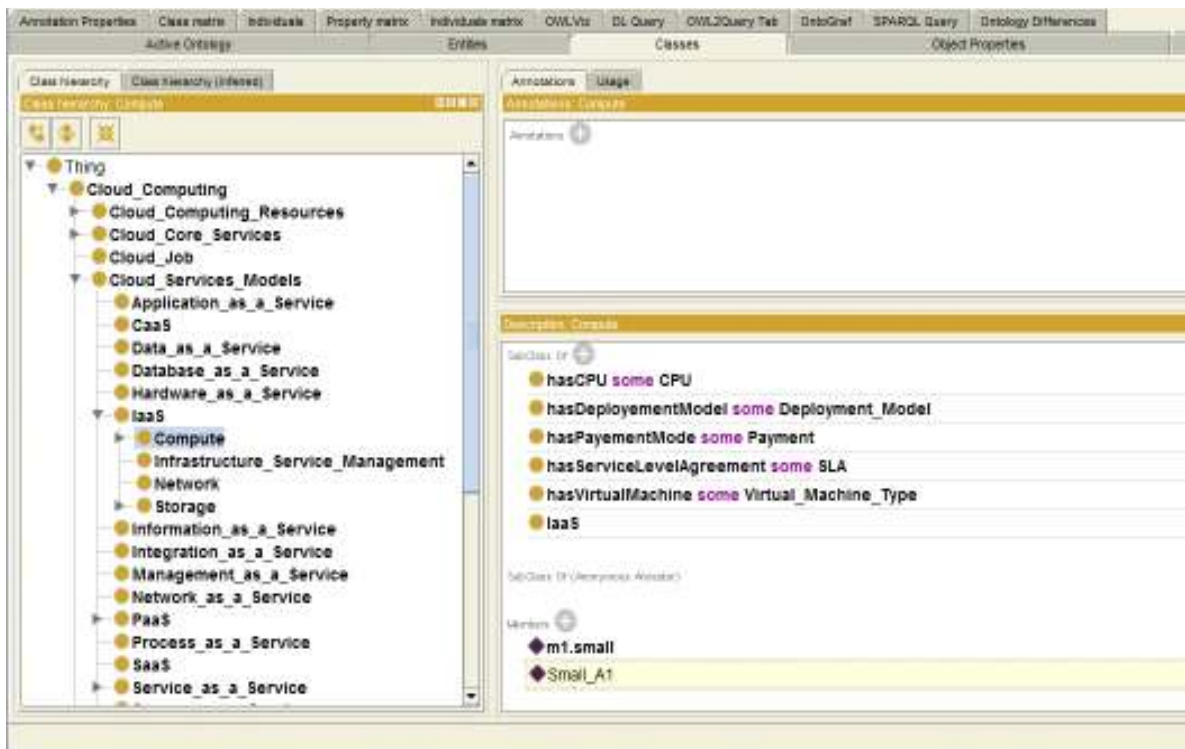Fig. 3: Example of cloud service instance



Fig. 4: Ontology classification returned by the reasoner

has supported a number of reasoners such as FaCT++ and HermiT. The FaCT++ reasoner has been used to assess the cloud computing ontology in this study. The reasoner is also very important for ontology query where the query is sent first to the reasoner for consistency checking versus the rules of the ontology. The query will be processed only if there are no inconsistencies, otherwise, the requester will receive the error message (Zeshan and Mohamad, 2012). The Reasoner checks if a specific class is a subclass of another class. If a class consists of any instances, it checks the consistency of the ontology and computes the inferred ontology class hierarchy (Horridge, 2009). In Protégé, it is possible to view the class hierarchy in two different modes: the asserted class hierarchy and the inferred class hierarchy. The asserted class hierarchy is the class hierarchy constructed manually by the developer before compiling the reasoner. The inferred class hierarchy is the hierarchy constructed by the reasoner. The result of the FaCT++ reasoner of this study is shown in Fig. 4.

## DISCUSSION

In cloud computing, many of cloud providers provide different cloud services with different attributes. Searching the cloud services with preferred attributes using the traditional search engines such as Google, Yahoo and others is not a suitable method due to the time consuming to browse and choose the

suitable service from the services returned offered by these search engines. Traditional search engines are based on the keyword search whereas searching for cloud services is multi-criteria based. To overcome the time consuming and keyword based searching issues (or limitations), knowledge-base system for representing the cloud computing services concepts in a semantic meaning is achieved using ontology. In information retrieval, service discovery, recommendations and other scopes, using ontology and semantic technology have been proved to be useful models (Rodríguez-García *et al.*, 2013).

In this study, an ontological model for organizing the knowledge of cloud computing services has been explored. The developed ontology explains the concepts, attributes of the concepts, axioms, individual and the relationship among these concepts for the cloud computing services domain. Table 3 shows the contribution of our work based on the following evaluation criteria.

**Taxonomy:** Is a hierarchy created according to data internal to the concepts in that hierarchy. It helps ontology engineers to find possible incompatibilities among the concepts.

**Consistency:** Means the ontology parts agree with each other. For example, the inferred knowledge from ontology does not contain contradictory knowledge.

Table 3: Work evaluation

| Ontology | Taxonomy | Consistency | Domain covered | Relationship | Preciseness | Automation |
|---|---|---|---|---|---|---|
| Rodríguez-García *et al.* (2013) | √ | √ | • Cloud services discovery<br>• Service models (SaaS, PaaS, IaaS) | √ | | Automatic |
| Youseff *et al.* (2008) | √ | | Service models (SaaS, PaaS, IaaS, CaaS, DaaS) | | | Manual |
| Weinhardt *et al.* (2009) | √ | | Service models (SaaS, PaaS, IaaS) | √ | | Manual |
| Dastjerdi *et al.* (2010) | √ | | • Service models (IaaS)<br>• Qos discovery | √ | | Manual |
| Ma *et al.* (2011) | √ | | • Resource management<br>• Service models (SaaS, PaaS, IaaS) | √ | | Manual |
| Moscato *et al.* (2011) | √ | √ | • Cloud services discovery<br>• Resource management<br>• Service models (SaaS, PaaS, IaaS)<br>• SLA | √ | | Manual |
| Kang and Sim (2011) | √ | √ | • Cloud services discovery<br>• Service models (SaaS, PaaS, IaaS) | √ | | Manual |
| Noor *et al.* (2013) | √ | √ | • Cloud services discovery<br>• Service models (SaaS, PaaS, IaaS) | √ | | Manual |
| Our ontology | √ | √ | • Cloud services discovery<br>• Resource management<br>• Service models (SaaS, PaaS, IaaS and others)<br>• SLA<br>• QoS | √ | √ | Manual |

**Domain coverage:** Specifies which domain of ontology is covered such as cloud services discovery and cloud computing resources management and which cloud services models support including SaaS, PaaS, IaaS and other models. In addition to the attributes of those cloud services such as QoS and SLA.

**Relationship:** Refers to the rich relations among concepts.

**Preciseness:** Means that the axioms, created relations and restrictions are involved in the constructed ontology.

**Automation:** Refers to the way of constructing the ontology where there are three kinds including manual, semi-automated and automated.

This study provides a comprehensive ontology that covers many aspects of cloud computing such as cloud services models (SaaS, PaaS, IaaS, DaaS, CaaS and others), cloud computing deployment (Private, Public, hybrid and community), QoS, SLA that enable the cloud computing researchers in different areas such as cloud services discovery. It also provides recommendations and resource management on how to benefit from this ontology.

**CONCLUSION AND RECOMMENDATIONS**

Cloud computing is a technology that provides everything as a service on-demand like public utilities (Water, Electricity, … etc.) over the Internet such as IaaS, paaS, SaaS and other services. Software Oriented Architecture (SOA) is an architectural approach that allows integrating and composite the application's components in a loosely coupled way during the development of the applications. Combining SOA and Cloud Computing have arisen the number of cloud providers which in consequence allows them to create different cloud computing services with different attributes and costs. Every cloud provider describes and represents the cloud services in its own format normally using HTML pages because there is no standardization for representing the cloud services. Searching cloud services represented in the heterogeneous concepts is a complex task. However, the ontology has been proved that it is useful for information retrieval, services discovery and other areas. In this study, a domain ontological model for representing the heterogeneous cloud computing services has been explored. We followed the METHONTOLOGY methodology during the constructing of this ontology and the Protégé tool is used to represent and check the consistency of this ontology. This developed ontology covers different aspects of cloud computing such as cloud services models, cloud computing deployment, service level agreements, QoS and others. The evaluation results show the feasibility and consistency of this ontology. It can be extended to involve the representation of API tools concepts which helps the developers to develop their applications on the cloud.

In future works, we plan to develop a system to help the cloud providers to represent the cloud services automatically via this system using ontology learning and Natural Language Processing (NLP) techniques.

**REFERENCES**

Ahmed, S., S. Kim and K.M. Wallace, 2007. A methodology for creating ontologies for engineering design. J. Comput. Inf. Sci. Eng., 7(2): 132-140.

Androcec, D., N. Vrcek and J. Seva, 2012. Cloud Computing ontologies: A systematic review. Proceeding of the 3rd International Conference on Models and Ontology-based Design of Protocols, Architectures and Services (MOPAS, 2012), pp: 9-14.

Bechhofer, S., F. Van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness and P.F. Patel-Schneider, 2004. OWL web ontology language reference. W3C recommendation, 10, 2006-2001.

Brickley, D. and R.V. Guha, 2000. Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation 27 March 2000.

Cardoso, J., A. Barros, N. May and U. Kylau, 2010. Towards a unified service description language for the internet of services: Requirements and first developments. Proceeding of IEEE International Conference on Services Computing (SCC), pp: 602-609.

Castells, P., M. Fernandez and D.Vallet, 2007. An adaptation of the vector-space model for ontology-based information retrieval. IEEE T. Knowl. Data En., 19(2): 261-272.

Chang, Y.S., T.Y. Juang, C.H. Chang and J.S. Yen, 2012. Integrating intelligent agent and ontology for services discovery on cloud environment. Proceeding of IEEE International Conference on Systems, Man and Cybernetics (SMC, 2012), pp: 3215-3220.

Chinnici, R., J.J. Moreau, A. Ryman and S. Weerawarana, 2007. Web services description language (wsdl) version 2.0 part 1: Core language. W3C Recommendation, 26.

Colomo-Palacios, R., Á. García-Crespo, P. Soto-Acosta, M. Ruano-Mayoral and D. Jiménez-López, 2010. A case analysis of semantic technologies for R&D intermediation information management. Int. J. Inform. Manage., 30(5): 465-469.

Dastjerdi, A.V., S.G.H. Tabatabaei and R. Buyya, 2010. An effective architecture for automated appliance management system applying ontology-based cloud discovery. Proceeding of 10th IEEE/ACM International Conference on the Cluster, Cloud and Grid Computing (CCGrid), pp: 104-112.

Fehling, C., T. Ewald, F. Leymann, M. Pauly, J. Rutschlin and D. Schumm, 2012. Capturing Cloud computing knowledge and experience in patterns. Proceeding of IEEE 5th International Conference on the Cloud Computing (CLOUD), pp: 726-733.

Fernández-López, M., A. Gómez-Pérez and N. Juristo, 1997. Methontology: From ontological art towards ontological engineering. AAAI Technical Report SS-97-06.

Flahive, A., B.O. Apduhan, J.W. Rahayu and D. Taniar, 2006. Large scale ontology tailoring and simulation in the semantic grid environment. Int. J. Metadata, Semantics Ontologies, 1(4): 265-281.

Flahive, A., D. Taniar, W. Rahayu and B.O. Apduhan, 2009. Ontology tailoring in the semantic grid. Comput. Stand. Inter., 31(5): 870-885.

García-Crespo, Á., J.L. López-Cuadrado, R. Colomo-Palacios, I. González-Carrasco and B. Ruiz-Mezcua, 2011. Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. Expert Syst. Appl., 38(10): 13310-13319.

Garg, S.K., S. Versteeg and R. Buyya, 2013. A framework for ranking of cloud computing services. Future Gener Comp. Sy., 29(4): 1012-1023.

Grüninger, M. and M.S. Fox, 1995. Methodology for the design and evaluation of ontologies. Proceeding of Workshop on Basic Ontological Issues in Knowledge Sharing.

Gruninger, M. and J. Lee, 2002. Ontology. Commun. ACM, 45(2): 39.

Han, T. and K.M. Sim, 2010. An ontology-enhanced cloud service discovery system. Proceeding of the International Multi Conference of Engineers and Computer Scientists, Hong Kong.

Han, T. and K.M. Sim, 2011. An agent-based cloud service discovery system that consults cloud ontology. In: Ao, S.I. *et al.*, (Eds.), Intelligent Control and Computer Engineering, pp: 203-216.

Han, S.M., M.M. Hassan, C.W. Yoon and E.N. Huh, 2009. Efficient service recommendation system for cloud computing market. Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp: 839-845.

Hausenblas, M., R. Grossman, A. Harth and P. Cudré-Mauroux, 2012. Large-scale Linked Data Processing-cloud Computing to the Rescue? CLOSER, pp: 246-251.

Hendler, J. and D.L. McGuinness, 2000. The DARPA agent markup language. IEEE Intell. Syst., 15(6): 67-73.

Hoefer, C.N. and G. Karagiannis, 2010. Taxonomy of cloud computing services. Proceeding of the 4th IEEE Workshop on enabling the Future Services-oriented Internet (EFSOI, 210), Workshop of IEEE GLOBECOM 2010 pp: 1345-1350.

Horridge, M., 2009. A Practical Guide to Building OWL Ontologies using Protégé 4 and CO-ODE Tools Edition1. 2. The University Of Manchester.

Jasper, R. and M. Uschold, 1999. A framework for understanding and classifying ontology applications. Proceeding of 12th International Workshop on Knowledge Acquisition, Modelling and Management (KAW, 1999), pp: 16-21.

John, 2009. Unified Ontology of Cloud Computing. Retrieved from: http://www.johnmwillis.com/cloud-computing/ unified-ontology- of-cloud-computing/ (Accessed on: 6/9/2013).

Kang, J. and K.M. Sim, 2010. Cloudle: A multi-criteria cloud service search engine. Proceeding of the IEEE Asia-Pacific Services Computing Conference (APSCC), pp: 339-346.

Kang, J. and K.M. Sim, 2011. Towards agents and ontology for cloud service discovery. Proceeding of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp: 483-490.

Kumara, S.R. 2006. A methodology for product family ontology development using formal concept analysis and web ontology language. J. Comput. Inf. Sci. Eng., 6: 103.

Lee, G., 2012. Resource allocation and scheduling in heterogeneous cloud environments. Ph.D. Thesis, University of California, Berkeley

Lupiani-Ruiz, E., I. GarcíA-Manotas, R. Valencia-GarcíA, F. GarcíA-SáNchez, D. Castellanos-Nieves and J.T. FernáNdez-Breis, 2011. Financial news semantic search engine. Expert Syst. Appl., 38(12): 15565-15572.

Ma, Y.B., S.H. Jang and J.S. Lee, 2011. Ontology-based resource management for cloud computing. In: N.T. Nguyen, C.G. Kim and A. Janiak (Eds.), ACIIDS 2011. LNAI 6592, Springer-Verlag, Berlin, Heidelberg, pp: 343-352.

Maedche, A. and S. Staab, 2001. Ontology learning for the semantic web. IEEE Intell. Syst., 16(2): 72-79.

Menzel, M. and R. Ranjan, 2012. CloudGenius: Decision support for web server cloud migration. Proceedings of the 21st International Conference on World Wide Web, pp: 979-988.

Miyano, T. and M. Uehara, 2012. Proposal for Cloud Search Engine as a Service. Proceeding of the 15th International Conference on Network-Based Information Systems (NBIS, 2012), pp: 627-632.

Moscato, F., R. Aversa, B. Di Martino, T. Fortis and V. Munteanu, 2011. An analysis of mOSAIC ontology for Cloud resources annotation. Proceeding of the Federated Conference on Computer Science and Information Systems (FedCSIS), pp: 973-980.

Noor, T.H., Q.Z. Sheng, A. Alfazi, A.H. Ngu and J. Law, 2013. CSCE: A crawler engine for cloud services discovery on the World Wide Web. Proceeding of the IEEE 20th International Conference on Web Services (ICWS, 2013), pp: 443-450.

Noy, N.F. and D.L. McGuinness, 2001. Ontology Development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems Laboratory, Stanford University.

Pearson, S. and A. Benameur, 2010. Privacy, security and trust issues arising from cloud computing. Proceeding of the IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom), pp: 693-702.

Ren, K., C. Wang and Q. Wang, 2012. Security challenges for the public cloud. IEEE Internet Comput., 16(1): 69-73.

Reshma, V.K. and B. Saravana Balaji, 2012. Cloud service publication and discovery using ontology. Int. J. Sci. Eng. Res., 3(5).

Rodríguez-García, M.Á., R. Valencia-García and F. García-Sánchez, 2013. Ontology-based annotation and retrieval of services in the cloud. Knowl-Based Syst., 55: 15-25.

Shih, W.C., C.T. Yang and S.S. Tseng, 2009. Ontology-based content organization and retrieval for SCORM-compliant teaching materials in data grids. Future Gener Comp. Sy., 25(6): 687-694.

Shimba, F., 2010. Cloud computing: Strategies for cloud computing adoption. Ph.D. Thesis, Dublin Institute of Technology, Dublin.

Sim, K., 2012. Agent-based cloud computing. IEEE T. Serv. Comput., 5(4): 564-577.

Smit, M., P. Pawluk, B. Simmons and M. Litoiu, 2012. A web service for cloud metadata. Proceeding of the IEEE 8th World Congress on Services (SERVICES), pp: 361-368.

Soundararajan, V. and K. Govil, 2010. Challenges in building scalable virtualized datacenter management. SIGOPS Oper. Syst. Rev., 44(4): 95-102.

Studer, R., V.R. Benjamins and D. Fensel, 1998). Knowledge engineering: Principles and methods. Data Knowl. Eng., 25: 161-199.

Tahamtan, A., S.A. Beheshti, A. Anjomshoaa and A.M. Tjoa, 2012. A cloud repository and discovery framework based on a unified business and cloud service ontology. Proceeding of the IEEE 8th World Congress on Services (SERVICES), pp: 203-210.

Truong, H.L., S. Dustdar and K. Bhattacharya, 2012. Programming hybrid services in the cloud. Proceeding of 10th International Conference on Service-Oriented Computing (ICSOC, 2012), pp: 96-110.

Valencia-García, R., F. García-Sánchez, D. Castellanos-Nieves and J.T. Fernández-Breis, 2011. OWLPath: An OWL ontology-guided query editor. IEEE T. Syst. Man Cy. A, 41(1): 121-136.

Valencia-García, R., J.T. Fernández-Breis, J.M. Ruiz-Martínez, F. García-Sánchez and R. Martínez-Béjar, 2008. A knowledge acquisition methodology to ontology construction for information retrieval from medical documents. Expert Syst., 25(3): 314-334.

Wang, C., J. Lu and G. Zhang, 2006. Integration of ontology data through learning instance matching. Proceeding of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp: 536-539.

Wei, Y. and M.B. Blake, 2010. Service-oriented computing and cloud computing: Challenges and opportunities. IEEE Internet Comput., 14(6): 72-75.

Wei, Y. and M.B. Blake, 2012. An agent-based services framework with adaptive monitoring in cloud environments. Proceeding of the IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE, 2012), pp: 4-9.

Weinhardt, C., A. Anandasivam, B. Blau and J. Stößer, 2009. Business models in the service world. IT Professional, 11(2): 28-33.

White, B., 2011. Accessibility challenges of the next decade: Cloud and mobile computing and beyond. Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, pp: 13.

Wu, J., L. Chen, Y. Xie and Z. Zheng, 2012. Titan: A system for effective web service discovery. Proceedings of the 21st International Conference Companion on World Wide Web, pp: 441-444.

Yongbo, J. and Z. Ruili, 2011. Intelligent search engines based on Web mining. Proceeding of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN, 2011), pp: 171-174.

Youseff, L., M. Butrico and D. Da Silva, 2008. Toward a unified ontology of cloud computing. Proceeding of the Grid Computing Environments Workshop (GCE'08), pp: 1-10.

ZadJabbari, B., P. Wongthongtham and F.K. Hussain, 2010. Ontology based approach in knowledge sharing measurement. J. Univ., Comput. Sci., 16(6): 956-982.

Zeng, C., X. Guo, W. Ou and D. Han, 2009. Cloud computing service composition and search based on semantic. Cloud Comput., pp: 290-300.

Zeshan, F. and R. Mohamad, 2012. Medical ontology in the dynamic healthcare environment. Procedia Comput. Sci., 10: 340-348.

Zhang, J., L.W. He, F.Y. Huang and B. Liu, 2013. Service discovery architecture applied in cloud computing environments. Appl. Mech. Mater., 241: 3177-3183.

Zhang, M., R. Ranjan, A. Haller, D. Georgakopoulos, M. Menzel and S. Nepal, 2012. An ontology-based system for cloud infrastructure services' discovery. Proceeding of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp: 524-530.

Zhao, L., Y. Ren, M. Li and K. Sakurai, 2012. Flexible service selection with user-specific QoS support in service-oriented architecture. J. Netw. Comput. Appl., 35(3): 962-973.