# A proposal for Infrastructure Cloud in Educational Environments

Jorge R. B. Garay*, Alexandre M. de Oliveira, Juan C. Z. Torres, Gustavo M. Calixto, Marcelo K. Zuffo

*Interdisciplinary Center in Interactive Technologies– CITI - Laboratory of Integrated Systems - LSI, Department of Electronic Systems Engineering, Polytechnic School, University of São Paulo, CEP: 05508-010 SP, Brazil*

**Abstract** - This paper describes a proposal for the development of an architecture for Educational Cloud based on the OpenStack framework, Service Composition techniques, and Semantic Web to create virtualized environments with specific features for teaching and researching environments. The creation and management of the architecture and its virtualized environment are done through a system of service composition. Services are made available as a catalogue and their access is limited by the user's profile type (student, teacher, researcher). The proposal has been approved and it is being developed in partnership with IBM Brazil. The results in this article are limited to the presentation and description of the architecture and modeling of the system's logic

**Keywords:** OpenStack; Service Composition; Cloud Computing; Semantic Web; SaaS, PaaS**.**

## 1. INTRODUCTION

Cloud Computing is a new concept or computational paradigm which allows offering computing services (applications, processing, storage, simulation etc.) over the Internet. In this paradigm, all the functionalities that may be provided for an information system are offered in the cloud as services, therefore, the final users can access the desired functionalities without having a deep understanding to build and manage them or maintain the required infrastructure (hardware) to run them as well (virtual machines, storage environments, processing, applications, etc.). In this scenario, both companies and educational institutions seek solutions that allow a cost reduction of their operations, especially IT-related ones. A solution that has been adopted by the private sector to reach that goal is the outsourcing of IT processes execution [1]. In an academic environment, though, part of the IT processes requires the creation of research and educational laboratory environments which meet the needs that come as a result of the analysis stages and development of the investigation and education work; however, the required resources, such as computers and specific software, might not be always available in these environments (laboratories), mostly due to lack of financial resources and funding.

*Address correspondence to this author at the Laboratory of Integrated Systems-LSI-Department of Electronic Systems Engineering, Polytechnic School, University of São Paulo, CEP 05508-010, São Paulo, SP, Brazil; Tel: +55 11 30919741; E-mail: jorge@lsi.usp.br

Students, teachers, and researchers need different development environments for learning, teaching, and researching purposes. However, those computational environments are not available due to their complexity regarding the implementation, to the high cost or to the bureaucracy for building and keeping dozens of different development environments, damaging the processes of research and education [2] among different centers of academic excellence in Brazil.

The concept of Cloud Application Platform as a Service (aPaaS) is quite recent [1] and presents a model where the development environments (application platform) are offered on demand. These development environments are stored in the cloud and are made available by following the cloud computing characteristics, sparing designers and application developers from reinventing the non-trivial wheel of setting up the development environment [3]. However, the use of aPaaS still requires skilled and specifically allotted labor for the creation management of the development environment as an application platform in the cloud [4].

Some of the techniques for resource management, such as Cloud service composition, have been shown as very efficient to promptly meet the needs of the users through solutions based on software agents [4] which allow to meet stages of negotiation between services. Other Cloud service composition solutions, such as described in [2], aim to integrate tests and service composition environments through the selection of implementation services over annotations, similar to the Google

Guice. In [5], a SOA (Service-Oriented Architecture) architecture model is used as a key element to the network composition, sorting services from the network infrastructure and the Cloud services, as well as developing techniques of analysis to determine the performance that can be delivered for composite network–Cloud services. To determine technique of service performance analysis managed by the Cloud, it demands an objective study of the non-functional aspects or requirements due to the increasing number of web services, always considering quality of service aspects [3] for selection of the most adequate method of Cloud Service Composition.

This work of research and development aims to develop an architecture for a system that allows creating and managing virtualized desktops that uniquely meets the needs of hardware and software from academic centers of teaching and research in Brazil. The work is developed in partnership with IBM Brazil and intends to exploit the potential of the OpenStack Framework combined with Cloud Service Composition and Web Semantics techniques.

The proposal aims to contribute to the availability and better management of virtual environments that meet the same features of the physical computers and servers, allowing the user to choose on demand among a finite number of virtual machine images, instrumentation equipment and applications for development and simulation. The goal is not restricted to creating an architecture of hardware and software to meet specific needs of centers of teaching and academic research but is also expected to develop measurable skills to motivate the use and creation of new applications on this proposal, which is currently under development.

This paper is organized in the following topics: It starts with a brief introduction to the main works, more related to our proposal. At section II, a detailed, but not less objective description of semantic web and service composition is inserted in the proposal. At section III, it is presented a complement of the proposal architecture detail. At section IV, it is presented an objective description of each function of the proposed architecture and, finally at section V, the conclusions about the proposal viability are presented.

## 2. SEMANTIC WEB AND SERVICE COMPOSITION

2.1. Semantic Web: The Semantic Web is defined as an extension of the current Web in which the meaning of the information is well-defined [10] and aims to provide a model capable to attribute a meaning to the contents generated by different data sources, like Web pages, database, sensors, and even information from our daily life, in a way they can be interpreted either by humans or machines (computers) automatically through software agents [10]. Thus, the Web Semantic formally describes an infrastructure for associating semantic descriptions to the information from the current Web under a representation that makes computers able to interpret and draw inferences from the data. The implementation of this infrastructure makes possible the generation of data recovery and the exchange and integration services, assuring data interoperability.

In order to use a Web service in the context of the Semantic Web, a software agent needs a semantic description of the service which defines the way the service can be accessed and executed. Thus, a clear infrastructure is needed to specify and share those semantic descriptions. A proposed solution for the structuring of semantic descriptions is the use of ontologies. One of the most quoted definitions in the literature on ontology in the context of Semantic Web is the one which defines "ontology as a formal specification, explicit and shared from a conceptualization" [11]. Thus, we can say that ontology is a description of concepts, relations and restrictions between concepts that can be used by a software agent to process information.

A Semantic Web service is nothing but a Web service which semantically auto-describes itself, therefore, it describes: its functionalities through the specification of restrictions (pre-conditions and post-conditions), and the way how it can be automatically found and invoked [12]. Among the ontological models that allow implementing this semantic description in the web service, we can highlight: OWL-S (Web Ontology Language for Services), WSMO (Web Services Modeling Ontology) and WSDL-S (Web Service Description Language for Services).

Using the Semantic Web infrastructure, it is possible to semantically annotate information about users, for instance, users with teacher, student and researcher profiles. In the proposal development, this semantic annotation must include information regarding preferences and competences in specific knowledge areas. In the same manner, information about pre-requirements (for your setup) and functionalities offered by specific software (in the Cloud) can be semantically annotated. Semantic Web Services can be developed in a way that the users can describe their needs on an specific software or development environment (which allows them to execute their academic assignments) in such a way that these service along the semantically annotated users information and software can be used by a system with the ability to search, select, compose and execute various tasks and processes aiming to build, almost automatically, a more suitable software infrastructure for each user, considering their preferences, skills, and restrictions on the software tool (licenses and complexity of use).

2.2. Service Composition: Usually, the composition is the reason for something to be decomposed. We divide something bigger in smaller parts because we identify the potential benefit of building things with the parts which could not be made if they were existing as a whole. These small parts

can be called by processes, which we sorted in three types for our architecture proposal: AS (Abstraction Service), CS (Composition Service), and TS (Translate Service).

***a) Abstraction Service;*** AS is one of the processes that interacts with the information available for and from the user, once it will allow us to distinguish it among abstract services generated by the system that concretely exist. In this process, the semantic web techniques and the use of ontologies as the main abstraction element are included, introducing a meaning for each service, easing and allowing, at the same time, the reuse and independent composition.

***b) Service Composition;*** SC defines the way of services composition. The services size and complexity tend to be modest. So, simpler configurations are called by primitive compositions and more sophisticated compositions are called by complex compositions. The main idea on the services composition for Educational Cloud is that as the proposed architecture corresponds to a specific scenario, the reuse of existing components must be larger or smaller, which would provide us a simplification aggregated level, independency, and the reuse during the composition process needed to obtain and provide a quick answer which is transparent to the requisitions that come from the same service by multiple users.

***c) Translate Service;*** TS, once the service composition is created, TS is used to describe the service composition in a service oriented language that allows its execution by the final user. TS optimizes the use of interfaces, drawing patterns along with the language used for translating the services. In some cases, specific languages can improve the expressivity of the available web services to the users through the interfaces, making them much more intuitive.

## 3. DESCRIPTION OF SERVICES FOR COMPOSITION IN THE CLOUD

On Figure 1, a set of services is presented, which some of them might affect direct or indirectly the level of reachable composition by a service inside the educational cloud (private). The described services below are a result from the adaptation of the workflow techniques, very common in service composition with SOA and Web Services, however, they are redefined for the cloud context, integrating service composition and semantic web.

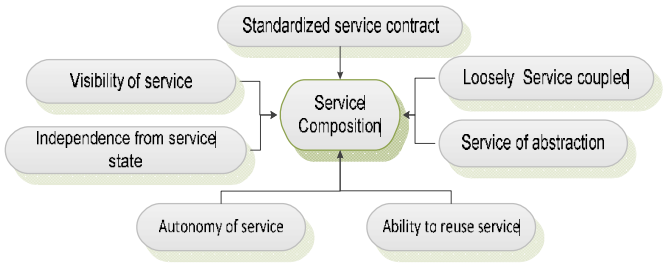


Fig. 1: Set of services defined for the implementation of the proposed architecture

***a) Standardized Service Contract:*** The contracts form a pre-established part of the system and every time that two services need to be linked, it will be necessary some form of technical standardized service contract [6]. In our case, a technical standardized service contract can be composed by a group of documents of services description, like WSDL or XML Shema.

***b) Loosely Service Coupled:*** The coupling represents only a relationship between two services or software components. In this process, there are various architectural options for design [7], by using middleware and intermediate elements, in our case only limited by the services characteristics to be supported by the OpenStack framework, the final goal is always to minimize the dependence between services.

***c) Service of Abstraction:*** It is here that the balance and regulation of other principles tendencies is done to add further definitions to the service contract, for this reason, we need to survey the values and the risks associated to the metadata from the publication services [8], minimizing, in the majority of the cases, the availability of metadata.

For the educational cloud service it would be highly beneficial to hide the details of the technology used to create a software program in order to keep the freedom to make technical modifications without affecting other users in runtime. Meanwhile, as a proposal to academic solution, it is expected to provide an open solution in order to favour the development of new specific solutions for the Educational Cloud environment.

***d) Ability to reuse service:*** There is a great difficulty in managing the reusability of services in some specific frameworks for the cloud, and on OpenStack the difficulty is not smoother. Summarizing, here is where the logic and the generic contract of the services are implemented.

***e) Autonomy of Service:*** The autonomy, comparing to the software, represents the independence in which a software can execute the service composition logic, aiming to improve the reliability and to predict the behaviour of the service. Thus, here lies the implementation of the functional limit of the service in runtime.

***f) Independence from service state:*** Here we highlight the need to reduce, or, in some cases, eliminate, the consumption of physical system resources as a result of the unnecessary processing that the virtualization of devices may demand in the cloud. The main goal is to maximize the scalability of a certain service, especially services with a higher probability to be reused and to take part in various compositions requested by the users of the educational cloud.

***g) Visibility of service:*** We implement communicative metadata to be searched in response to the directed consults (Figure 2) by the selection criteria [9] arranged in the educational cloud services catalogue.

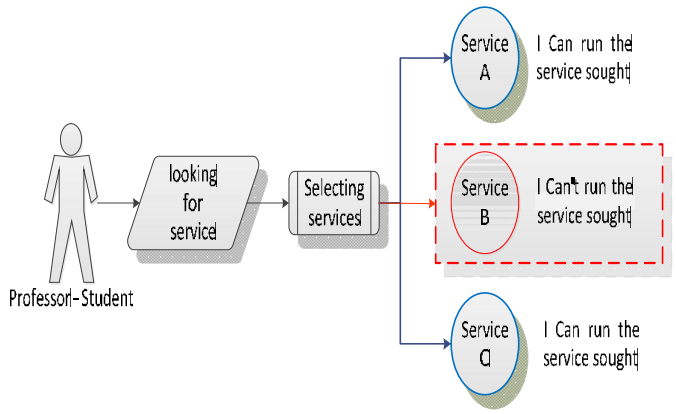


Fig. 2: Visualization of service process by the user consists in consult and filter

## 4. ARCHITECTURE DESCRIPTION

There is currently an increasing number of technologies for cloud computing, such as Eucalyptus, OpenNebula, Nimbus, AbiCloud [13][14], which are classified as open source and can be easily inserted in the context of our proposal, however, the adopted technology corresponds to the OpenStack framework. The proposal of using this OpenStack framework [14] is a bet from IBM due to the fast growth of the framework in both open source and functionalities. The proposal describes the possibility to provide to the user (teacher or student) virtualized environments with a pre-defined architecture, exploring the use of open source tools in order to build a cloud infrastructure as a service of specific use to the educational cloud environment.

In this context, and incorporating the semantic web, the proposal to be developed over an IBM Cloud architecture, aims to provide to the user (teacher, student and researcher) the ability to describe formally: their software needs (software development environment needed to perform their teaching or research activities) and their knowledge and preferences about certain software, so that an automatic composition (service composition techniques) can search in a software repository (Catalogue) for the proper tools to build the development environment required by the user.

In the interface module (Architecture - Figure 3), the user can specify explicitly or implicitly their software and hardware requirements (measuring equipment such as spectrum analyzers, oscilloscope etc., virtualized) or development environment needed for their teaching or research activities. The explicit description happens when the user indicates specific software, such as the Eclipse IDE for Java. Meanwhile, the implicit description comes when the user sets, for example, the need for some IDE that allows JAVA programming, in which some possible solutions would be IDEs Eclipse, NetBeans etc.

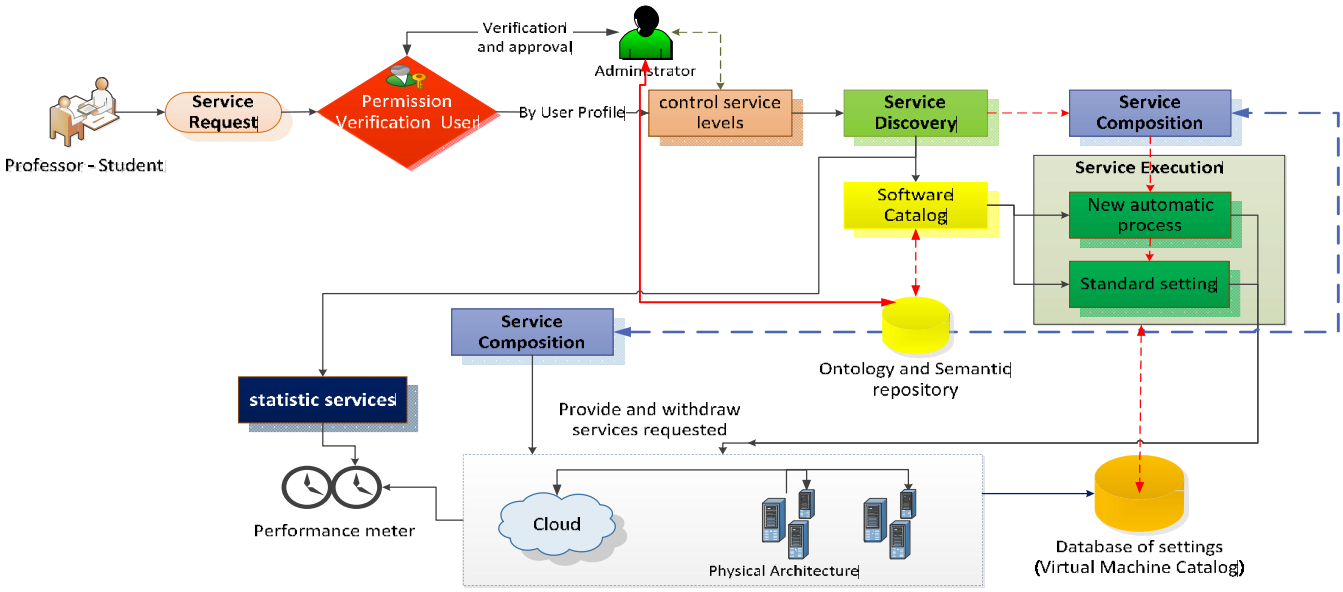


Fig. 3: Distributed Architecture for Educational Cloud

## 5. FUNCTIONS OF ARCHITECTURE

The proposed architecture must mostly develop skills to use the services of the Educational Cloud and motivate the development of new specific applications of the teaching and research environment. Then, it must also control the private and specific resources, create access interfaces that are more intuitive to services through the use of semantic web, allowing end users to access instrumentation (oscilloscope, spectrum analyzer etc., available through the cloud), supplying and optimizing resource utilization, streamlining the availability of computational and storage resources and finally the optimization of services with the highest demand.

Below, the relevant functionalities for the interaction between user and educational cloud are described in a succinct way:

***a) Service Request:*** allows users to describe their requirements for computing resources in a quick manner and without contacting the cloud administrator.

***b) Users control****:* by the own cloud characteristic, the existence of a user control mechanism allows to keep a profile and their most common functions in the academic environment. For instance, students and teachers' profiles and functions. The attribution of profiles and functions to new users can be done at the registering process.

***c) Level control service:*** a user (student, for example) can have a profile attached which does not have enough permission to access some functionality that is part of the solution he has required. In this case, the system administrator can verify the access request to the further levels (access to specific software or license restricted), changing the user's profile and functions.

***d) Service Discovery:*** the request sent by the user is sought in the available software catalogue. Because the user's request may be either implicit or explicit, the Service Discovery must uses the ontologies repository to determine if there is any software in the catalogue that can satisfy the user's request. When the request is satisfied, the component statistic service is called. Otherwise, the user's request is sent to the service composition.

- *Software catalogue:* it corresponds to the repository where the software executable files are stored in the educational cloud. The software catalogue also includes access to virtual machines previously created by the composition process and also access to measuring instruments available through the cloud.
- *Ontology repository:* it allows describing, with no ambiguity, information about the software present in the software catalogue. This information includes data about hardware and software pre-requirements and licensing for certain software. Besides that, the ontologies are used to understand (distinguish) the user's request when it was implicitly described.

***e) Service Composition:*** it is called when the service discovery does not find a service that can satisfy the user's request in the software catalogue. The service composition uses the ontology repository to search services that, composed in a specific order and configurations, allow satisfying the user's request. Because the services represent software, these services also specify dependencies information, such as some software which needs the previous installation of other software, or they run only on specific hardware platform or operating system. Beside that, the access or need for licenses is a pre-condition handled at the composition process. As a result, the service composition generates a script, indicating the software elements needed to build the virtual machine that can satisfy the development environment requested by the user.

***f) Service Execution:*** the script generated by the service composition is used along other OpenStack tools in order to build the virtual environment that represents the desktop environment requested by the user. In parallel, the service execution registers the new created resource in the software catalogue and requires to the system administrator to add the information about the new resource into the ontology repository, in such a way that it can be found or handled without ambiguities at eventual discovery processes or composition.

***g) Statistic service:*** this component allows creating a statistic model over the use of hardware and software resources, in such a way that the information can be analysed aiming to improve the load processing balancing, optimizing the cloud performance, as well as that information is used to analyse profiles, access levels and resource consumption by the users.

***h) Educational cloud:*** it represents the infrastructure based on specific features of the teaching and research environment (disciplines, demand for skills as well as associated hardware and software resources).

Figure 4 represents the first user interface attempt, in which the user's request can be explicitly specified. This interface requests pre-validation of user login (teacher or student). The explicit specification of user's request is done through a simple classification (using concepts relationships on ontologies) of the provided functionalities. The functionalities classification is done through categories and, by selecting one option (Matlab, for example), a new screen will be presented, showing the necessary elements (other necessary softwares) to build a virtualized desktop environment.

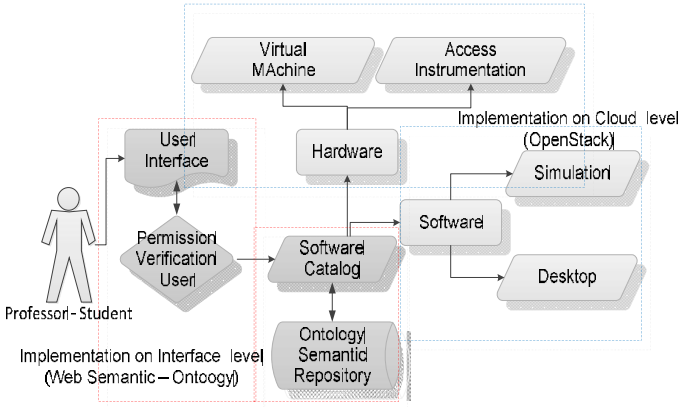


Fig. 4: Diagram - Catalogue Interface for Service Composition

## 6. CONCLUSIONS

The objective of our proposal is to describe an architecture to develop a composition system which ease the implementation of virtualized desktop environments in such a way that the needs for software and hardware infrastructure for the academic

and scientific community are supplied from the formal specification of these needs. To reach our goal, new large scale resource sharing and management technologies based on the concept of Cloud Computing were explored, which, when aimed to an academic environment, are called by Educational Cloud Services.

The proposal aims to describe, in a concise way, the components and their interaction for a composition system oriented to build virtualized desktop environments that are specific to the teaching and research environment. The proposed architecture must also develop skills to use the services of the Educational Cloud and motivate the development of new specific applications of the teaching environment.

The system is described as a system with horizontal distributed resources, shown to the Educational Cloud users as virtual resources. The result of the composition process is a virtualized desktop environment that contains software tools and hardware characteristics defined by the user. The construction of these virtualized environments is based on a plan generated by the composition system, implemented using open source tools, such as the OpenStack framework, so they can promptly be available as an Application Platform as a Service (aPaaS).

The use of Semantic Web concepts is applied in the process of clear description of the software and hardware requirements by the user (service request interface). Besides that, the Semantic Web can also allow modelling information over the user (profile, preferences etc.) and over the software tools (licenses, software, hardware pre-requirements etc.), in such a way that this semantic improvement allows the construction of more intuitive interfaces or aware of the context, which means that the preferences are imported from a user profile (semantic and contextually annotated) to help the process of inference and composition. The semantic annotation will initially depend on the application characteristics, allowing the use of ontological languages and models such as WSML/WSMO or OWL-S, and framework with Semantic Web programming support such as JENA. The proposal aims to ease the creation, management and maintenance processes of virtualized desktop environments, which results in a small investment at the beginning of research projects, as [15][16]. The implementation processes of the virtualized desktop environments turn to be much faster and simpler, bringing less risks, reducing bureaucracy, considering licenses management rules, and allowing an environment personalization level in function of the users' competences and skills.

## REFERENCES


[1]. R. Mietzner; F. Leymann; , "A self-service portal for service-based applications," Service-Oriented Compcriauting and Applications (SOCA), 2010 IEEE International Conference on , vol., no., pp.1-8, 13-15 Dec. 2010.

[2]. Wei-Tek Tsai; Peide Zhong; J. Balasooriya; Yinong Chen; Xiaoying Bai; J. Elston; , "An Approach for Service Composition and Testing for Cloud Computing," Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on , vol., no., pp.631-636, 23-27 March 2011. doi: 10.1109/ISADS.2011.90.

[3]. Huihui Bao; Wanchun Dou; , "A QoS-Aware Service Selection Method for Cloud Service Composition," Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International , vol., no., pp.2254-2261, 21-25 May 2012.

[4]. J.O. Gutierrez-Garcia; Kwang-Mong Sim; , "Self-Organizing Agents for Service Composition in Cloud Computing," Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on , vol., no., pp.59-66, Nov. 30 2010-Dec. 3 2010.

[5]. Qiang Duan; , "Modeling and Performance Analysis on Network Virtualization for Composite Network-Cloud Service Provisioning," Services (SERVICES), 2011 IEEE World Congress on , vol., no., pp.548-555, 4-9 July 2011.

[6]. C. Mauro; Sunyaev, A.; Leimeister, J.M.; Krcmar, H., "Standardized Device Services - A Design Pattern for Service Oriented Integration of Medical Devices," System Sciences (HICSS), 2010 43rd Hawaii International Conference on , vol., no., pp.1,10, 5-8 Jan. 2010.

[7]. D. Linner; I. Radusch; S. Steglich; C. Jacob, "The Semantic Data Space for Loosely Coupled Service Provisioning," Autonomous Decentralized Systems, 2007. ISADS '07. Eighth International Symposium on , vol., no., pp.97,104, 21-23 March 2007

[8]. Binh Minh Nguyen; Viet Tran; L. Hluchy, "Abstraction approach for developing and delivering cloud-based services," Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on , vol., no., pp.1,6, 18-20 Dec. 2012

[9]. D. Biswas; K. Vidyasankar, "Spheres of visibility[Web services]," Web Services, 2005. ECOWS 2005. Third IEEE European Conference on , vol., no., pp.12 pp.,, 14-16 Nov. 2005. doi: 10.1109/ECOWS.2005.23.

[10]. T. Berners-Lee, James Hendler, and Ora Lassila. "The semantic web." Scientific american 284.5 (2001): 28-37.

[11]. R. Gruber. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2 (1993): 199-220.

[12]. J.J. Chahoud. Planejamento para serviços Web semânticos. Dissertação de Mestrado. Instituto de Matematica de Estadistica da Universidade de São Paulo. 2006.

[13]. M. Mahjoub; A. Mdhaffar; R.B Halima; M. Jmaiel. "A Comparative Study of the Current Cloud Computing Technologies and Offers," Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on , vol., no., pp.131-134, 21-23 Nov. 2011. doi: 10.1109/NCCA.2011.28.

[14]. Xiaolong Wen; Genqiang Gu; Qingchun Li; Yun Gao; Xuejie Zhang; "Comparison of open-source cloud management platforms: OpenStack and OpenNebula," Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on , vol., no., pp.2457-2461, 29-31 May 2012. doi: 10.1109/FSKD.2012.6234218.

[15]. J.R.B. Garay; J.A.A. Palacio; A.M. de Oliveira; S.T. kofuji; S.I. B. "Optical Character Recognition and Zigbee Wireless Communication for AMR: Conclusive Project". Recent Patents in Engineering, v. 7, p. 125-132, 2013.

[16]. A.M. de Oliveira; J.R.B. Garay; J.F. Justo; S.T. Kofuji . A Complete Ultra Wide Band Transmitter System for Impulse Radar. Recent Patents in Engineering, v. 7, p. 133-139, 2013.