Seattle
Making Value for Society

122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Paper ID #13848

Work-in-Progress: Leveraging Cloud Computing and Web Standards to Support Learning Objectives in Multiple Classrooms

Ms. Amber Shanice Solomon, Clemson University
Dr. Sekou L Remy, Clemson University

Sekou L. Remy is a researcher focused on removing barriers to effective use of Robotics and Cloud Computing in our homes, schools, and training centers. Dr. Remy is currently an Assistant Professor in Human-Centered Computing, and comes to Clemson from the University of Notre Dame where he was a Moreau Postdoctoral Fellow. He also had the pleasure of serving as a part-time instructor in Computer Science at Spelman College. A graduate of the Georgia Institute of Technology (ECE) and Morehouse College (CS), Remy leverages education in both engineering and liberal arts to enable change.

Work-in-Progress: Leveraging Cloud Computing and Web Standards to Support Learning Objectives in Multiple Classrooms

Abstract

Cloud Computing is one of the newer technological trends that will have a significant impact on teaching and the learning environment[5]. Currently, Cloud Computing is not fully used to its potential to support learning. In this work we present cases that highlight how the Cloud Computing framework, including PaaS, IaaS, and web technologies were used to directly support learning objectives in five courses.

1. Introduction

According to the National Institute of Standards and Technology (NIST), Cloud Computing is, "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources." These resources are designed to be managed quickly and easily with minimal effort by either the provider or by the user. Such an approach to computing is broadly of interest, but while the potential application in education is exciting, it does provide some cause for concern. If change is implemented improperly, the impact on learning could be catastrophic. At the same time, if these technologies are not actively studied, the opportunities that they can afford can also be missed, and that too can be catastrophic given significant needs in workforce development for the new knowledge economy.

In 2013, CDW Government released a study that indicated that storage, collaboration/communication and office/productivity were the top three applications of cloud technologies applied in K-12 education[3 12]. These applications had a staggering penetration of 40% for storage, 36% for collaboration/communication, and 33% for office/productivity tools. Product offerings from Microsoft and Cisco indicate clear similar shifts that are anticipated in the academy as well[2 6]. What is not yet happening, however, is widespread use of other Cloud Computing service models like Infrastructure as a Service (IaaS) or Platform as a Service (PaaS) in the classroom to support learning objectives directly. The XSEDE Cloud Survey Report[9] presents a view of the ways that educators, students, and practitioners envision the support that cloud technologies can provide. What we believe is missing from the list of good candidates for the cloud (Finding #2), is the use of these cloud frameworks to directly support learning. We consider our approach as an additional pathway for educators and researchers to leverage the cloud (Finding #1). To confirm this view however, an IRB approved survey will be conducted in the future.

In this work we present cases that highlight how PaaS, IaaS, and web technologies like Browsers and Web Servers can be used to directly support learning objectives in five classes taught by faculty in the School of Computing at Clemson University. Beyond presentation of the technology and its applicability to support specific objectives, the broader work also documents sentiment about its use, from a student and faculty perspective. The technologies used, and the resources created, have strong applicability and can be extended to courses in Mechanics of Mechanisms, Control Theory, Artificial Intelligence, and Bioinformatics.

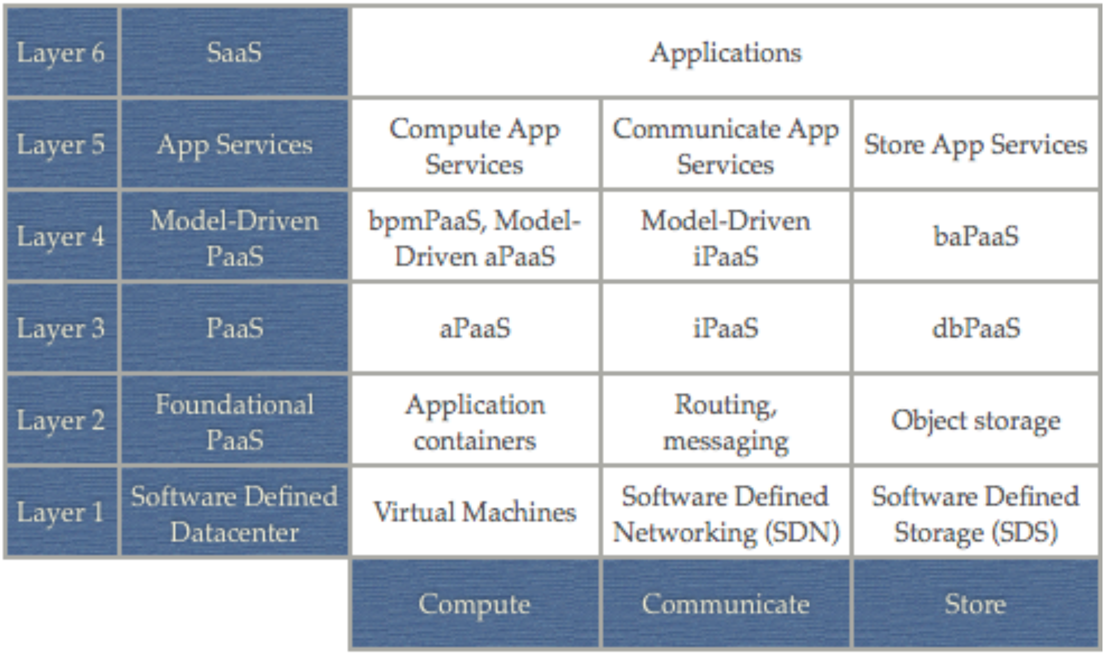
2. Background

Cloud computing is an attractive technology due to three main factors: its dynamic scalability, its ability to be used when the availability of resources is limited, and it requiring no specialized knowledge to use[5 7]. Because of these attractions, Cloud Computing can have a significant impact on the teaching and learning environment[7]. Current research tends to focus more on its applications, including using the technology for future distance education, teaching information systems, integrating teaching resources, and teaching systems development[5].

Cloud Computing applications including different operating systems (e.g. Ubuntu, Linux, Windows), web applications (e.g. Koding.com, Cloud 9, Brython, and Python Anywhere), and web services (e.g. webpy and mongoose) have applicability in educational settings. In further sections we explore the possibilities these technologies, and others like them, could have on education in the classroom.

2.1. Overview of the tools

The framework of the Cloud Computing platform can be divided into six layers[4] shown in Figure 1. For our purposes, we will focus on layers one, three, and six: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS)[4 5 7]. These layers were chosen because they are widely accessible forms of Cloud Computing, and they are not broadly leveraged in the classroom to directly support learning objectives. Some of the other resources are used to facilitate student collaboration, or to support classroom administration, or other more abstract learning activities, but we seek to highlight new opportunities.



Layer 6	SaaS	Applications		
Layer 5	App Services	Compute App Services	Communicate App Services	Store App Services
Layer 4	Model-Driven PaaS	bpmPaaS, Model-Driven aPaaS	Model-Driven iPaaS	baPaaS
Layer 3	PaaS	aPaaS	iPaaS	dbPaaS
Layer 2	Foundational PaaS	Application containers	Routing, messaging	Object storage
Layer 1	Software Defined Datacenter	Virtual Machines	Software Defined Networking (SDN)	Software Defined Storage (SDS)
		Compute	Communicate	Store

Figure 1. Den Haan's framework to categorize and compare cloud platforms[4].

2.1.1. Layer 1 - Infrastructure as a Service (IaaS)

The IaaS layer, referred to as Software Defined Datacenter by Den Haan, is mainly used for sharing computing resources. Simply stated, this layer allows users to deploy and run software. This layer ensures scalability and efficient use of resources for the Cloud Computing platform. It includes the hardware, storage, operating system, and resource pool. It uses hardware and software virtualization technology to ensure stability and reliability of the infrastructure. Virtualization allows features like caching and high availability to be independent from the hardware and allows more optimizations to lower performance degradation. Examples of applications for this layer include the different operating systems/virtual machines students may use in a learning environment, including Ubuntu, Linux, and Windows[4].

2.1.2. Layer 3 - Platform as a Service (PaaS)

This layer is focused on code. Here, code is defined as, "text that describes an application in a certain programming language, integration configurations, or database configurations[4]." In other words, code can be deployed and runtime packages are already available as a service. This layer allows communication amongst applications, as shown in the column iPaaS (integration platform as a service). The dbPaas column (database platform as a service) allows availability and scalability. PaaS also provides standard interfaces and the content of an API for the higher layers. This layer includes web services like web.py and mongoose. These applications are web frameworks or libraries that allow development of Python or C, respectively, web applications[13].

2.1.3. Layer 6 - Software as a Service (SaaS)

As we move up in layers, each layer abstracts from technical details until we reach actual applications[4]. SaaS focuses on applications and can be built on previous layers. This layer requires applications to compute, communicate, and store data. Examples of SaaS include web browser applications like Brython, Koding.com, Cloud 9, and Python Anywhere. These applications allow users to program and compile in the browser[1]. Using SaaS applications makes them accessible from various devices.

2.2. Overview of the expected impact

There are several advantages of Cloud Computing: powerful computing and storage capacity, high availability, high security, and virtualization. The major advantage is it provides easy access to software and does not require specialized knowledge to use, making it a great benefit for teachers in classrooms.

In the standard classroom, professors conduct lectures, train students in a skill, and provide work assignments and feedback on those assignments. The students generally work alone, occasionally interacting with peers and teachers[7]. However, with the use of PaaS, IaaS, and web technologies students could interact with teachers more regularly. Furthermore, these technologies can allow teachers to use multimedia to enhance their teaching content. These applications can provide easier and more efficient ways for teachers to monitor a student's progress[7].

3. Intended Learning Outcomes

Students are more motivated and will obtain a greater understanding when they see how a concept applies in a context. Therefore, students must have examples to understand complex ideas[11]. The layers of the Cloud Computing framework can be used to support learning objectives in different classes. The courses discussed in this section demonstrated the applicability and possibility of different Cloud Computing applications. Many of these classes were undergraduate computer science courses, with the exception of one undergraduate and graduate computer science course. Most courses were only populated with computer science majors; however, there was one multidisciplinary class. The applications to be described were first demonstrated by teachers in the classroom. Students were then assigned take-home projects, where they could collaborate with others and use the technology.

3.1. Mobile Software Development

In this undergraduate course, computer science majors developed mobile Android or iOS apps of their choice. This autonomous class expected students to be proficient programmers. To support this course, students leveraged IaaS for a semester long project.

The learning objectives for this class are as follows:

1. To understand the event-driven programming model as used in mobile applications.
2. To learn and use the interactive development environment and frameworks used in developing mobile applications for the Android operating system.
3. To critically evaluate alternate mobile application designs.
4. To learn how to develop applications using one or more of the tools available on mobile devices (such as the camera, GPS receiver, phone, Internet browser, a backend database, accelerometer, and geotagging).
5. To propose, design, and implement a non-trivial semester project.
6. To formally present and demonstrate the results of the semester project in a class mini-conference at the end of the semester.

Students developed an application, the ROS Robot Remote (R3), which used the unique iOS hardware to provide a remote control for robots. In that course, students used IaaS to simulate virtual robots, robots that provided APIs to access sensor data and modify supported actuators. The API used to interface with the simulated robot was also the same used to with a hardware robot. The use of the IaaS hosted simulation enabled more flexible testing and development, and also provided access to multiple simulations on demand, and as needed. The developed application provided an iOS user with an interface to control robots within a virtual world. The application used the gyroscope and motion sensors of iOS devices to control the robot.

3.2. Robotic Simulation and Modeling

In this course students explore useful topics in robotic simulation. To support this course, students leveraged IaaS and PaaS in two of the three projects in the semester. Their emphasis is on the tools and approaches that can be used to perform an experiment. Specific focus was applied on Open Source tools. The broad goal of the course is for students, who have some programming skills, to develop implementations in multiple types of simulations.

The learning objectives for this class are as follows:

1. To be conversant about the state of the art in Open Source robotic simulation.
2. To competently describe roles of significant tools involved in robotic simulation.
3. To demonstrate tangible application of simulation tools in an experiment.
4. To be able to identify which (if any) of the presented resources are of merit to their career aspirations or personal research.

In this multidisciplinary class, mechanical engineering and computer science undergraduates worked with computer science, Human-Centered Computing, and electrical engineering graduate students. Instead of using actual robots, students used IaaS to simulate robots. The use of IaaS required little to no knowledge and comprehension of control theory. Students were able to use HTTP get request to issue commands through the browser to get data to the robot to stimulate the robot's movements. They further programmed a robot's actions using PaaS, with languages including Java, C, and Matlab.

3.3. Creative Inquiry: Bio-inspired heuristics for solving large-scale optimization problems

In this course students analyze multiple heuristic strategies for optimization. Implementing the algorithms, students also develop a framework to permit head-to-head comparison of the implemented approaches.

The learning objectives for this class are as follows:

1. To develop new heuristic algorithms for solving challenging optimization problems
2. To develop visualizations and other educational tools to make existing heuristics easier for students and practitioners to understand at an intuitive level
3. To apply heuristics to real-world optimization problems facing the University and community.

Again, IaaS was used to simulate robots. None of the undergraduate computer science students had knowledge of control theory. They still, however, were able to use HTTP get request to issue commands through the browser to simulate the robot's movement and to get data to the robot. Students were also able to use IaaS, PaaS, and SaaS to optimize algorithms.

3.4. Computer Programming I

In this introductory programming class, special emphasis is placed on algorithm development and software life cycle concepts. This course is an introduction to modern problem solving and programming methods using C. Students use appropriate tools and discuss ethical issues arising from the impact of computing upon society. Topics in this course include Unix; representation of integer, floating point, and character data; declarations; loops and conditionals; functions and parameter passing; searching and sorting; structured data types; and pointers. The students used IaaS to implement one of the three programming assignments during the course.

The learning objectives for this class are as follows:

1. To demonstrate understanding of the basic problem solving process of implementing a computer program to solve a problem.
2. To write and run simple computer programs using C
3. To appreciate the need of modular programming to solve bigger and more complicated programs.
4. To remain excited about computing and its application to problem solving in application domains.

Although an introductory course, undergraduate computer science students were able to program a robot to follow a defined path. Students used IaaS to issue commands and get data using curl HTTP get request. Although, students were unaware they were using curl HTTP get request because the command was included in a header file that they were told to include in their assignment. Students received feedback of the robot's location by using a robot logger. They also had access to an onboard camera from the robot, so they had an egocentric view of their work.

3.5. Fundamentals of Robotics and Soft Computing

In this course students explore useful topics in robotics and soft computing. Their emphasis is on the tools and approaches that can be used to implement a "robotic brain" (high level control). The broad goal of the course is for undergraduate computer science students, who have some programming skills, to develop the capability to convert sensing and actuation into resources for assistive robotics. Students will complete projects that apply the algorithms described in research articles to the themed projects for the course.

The learning objectives for this class are as follows:

1. To be conversant about the state of the art in robotics and intelligent machines.
2. To competently describe roles of significant tools involved in robotic software development.
3. To demonstrate tangible application of soft computing techniques to robotics.
4. To discuss in detail how these techniques can be applied to help facilitate human operation
5. To be able to identify which (if any) of the presented methods are of merit to their career aspirations or personal research.

In this class students used IaaS to issue commands using HTTP get request. The students built their applications using PaaS and C and Python languages. They demonstrated their work using applications on SaaS.

4. Tools

Many of the classes used the same web technologies and web services to support learning. Web services refer to the technologies that allow making connections between web applications[11].

4.1. IaaS

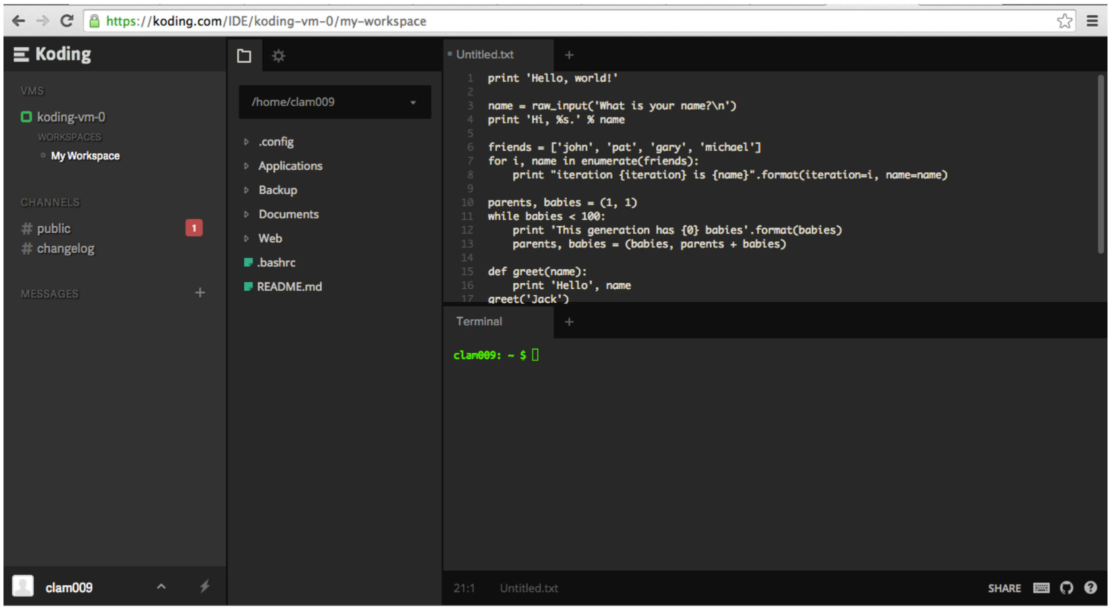
IaaS is composed of information infrastructure and teaching resources[5]. This layer was the most universal, used in all classes. Using this layer allowed students to make HTTP get requests to issue commands and to get data. Although, in some cases, students had little to no understanding of control theory, they were still able to successfully control a robot’s actions. Therefore, IaaS could be especially useful in the classroom because it can allow students to implement complex theories without the student having previous knowledge of the theory. IaaS was used in the same manner, however dependent upon the class and context, that manner taught different learning objectives. Because IaaS supports real-time simulation, this layer provided immediate feedback when a student changed his or her code. This was illustrated by the feedback from the robot logger.

4.2. PaaS

PaaS is mainly composed of the operating system and middleware. Combined with a browser, applications on this layer, like web.py, can further support learning and comprehension by allowing collaboration; team members can work and program remotely. PaaS can be especially useful for dynamic system optimization and modeling[5]. PaaS was used in the Brain CI course to allow students to optimize algorithms to control a robot.

4.3. SaaS

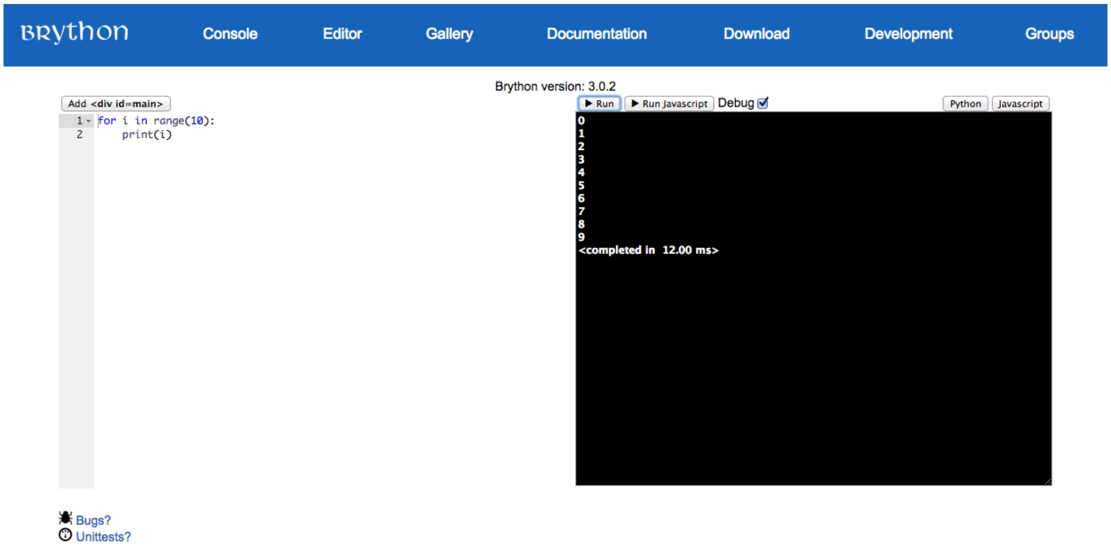
SaaS is composed of applications and user interfaces. This layer could be useful for teachers by allowing the sharing of teaching material resources[7]. SaaS applications like Brython, Koding.com, Cloud9, and Python Anywhere can support learning objectives by making abstract, complex ideas visual. They can be used for debugging, troubleshooting, and introspection by allowing students to change their code and immediately see the impact their changes made[7]. In our example, the web browser presented a visual representation of the robot in the Computer Programming I course. Figure 3 is an image of the Koding.com web interface, while Figure 4 is an image of the Brython web interface[1] [8].



```
print 'Hello, world!'

name = raw_input('What is your name?\n')
print 'Hi, %s.' % name

friends = ['john', 'pat', 'gary', 'michael']
for i, name in enumerate(friends):
    print "iteration {iteration} is {name}".format(iteration=i, name=name)

parents, babies = (1, 1)
while babies < 100:
    print 'This generation has {0} babies'.format(babies)
    parents, babies = (babies, parents + babies)

def greet(name):
    print 'Hello', name
greet('Jack')
```

```
clam009: ~ $ 
```

Figure 3. Koding.com web interface.



```
for i in range(10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
<completed in  12.00 ms>
```

Figure 4. Brython web interface.

5. Evaluations

Teachers suggested these technologies and tools could be very useful in the classroom and for teaching. Currently, many teachers use technology, such as BASH Shell, to support their learning objectives. In this project we seek to catalog the technologies that teachers can use both at research intensive and teaching intensive institutions. We also seek to formally assess how the presented works may be incorporated into their existing practice. Ultimately, our goal is to develop effective solutions that make use of the newer cloud technologies where it is reasonable for them to be used. By engaging educators and students in the design process we expect to have success developing usable, engaging applications that support users in the classroom. As an added benefit, the applicability of the web and cloud services also bears promise to greatly enhance the student's learning outside of the classroom as well.

6. Conclusions

Cloud Computing has emerged as a paradigm for managing and delivering services over the Internet and could have tremendous affects on learning[5]. The advantage of the Cloud Computing framework is the possibility of utilizing the same technologies to support different learning objectives.

Three layers in the Cloud Computing framework are of most interest: IaaS, PaaS, and SaaS. These applications were used to directly support learning objectives in five classes. Many of the classes utilized the same technologies to support learning different learning objectives. IaaS allowed students to make HTTP get requests to issue commands and to get data. PaaS allowed students to optimize algorithms to control a robot. SaaS applications can be used for debugging, troubleshooting, and introspection by allowing the student to change the code and immediately see the impact their changes made.

7. References

[1] Brython. Retrieved February 2, 2015, from http://brython.info/

[2] Cloud Computing in Education - Industry Solutions. Retrieved February 2, 2015, from http://www.cisco.com/web/strategy/education/cloud_computing.html

[3] Cloud Computing To Make Up 35% of K-12 IT Budgets in 4 Years -- THE Journal. Retrieved February 2, 2015, from http://disq.us/8kky03

[4] Den Haan, J. 2013. The cloud landscape described, categorized, and compared - by Johan Den Haan. Retrieved February 1, 2015, from http://www.theenterprisearchitect.eu/blog/2013/10/12/the-cloud-landscape-described-categorized-and-compared/

[5] Dong, B., Zheng, Q., Yang, J., Li, H., Qiao, M. 2009. An E-learning Ecosystem Based on Cloud Computing Infrastructure. *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on* , vol., no., pp.125,127, 15-17.

[6] Education Solutions. Retrieved February 2, 2015, from http://microsoft.com/education/en-us/solutions/Pages/cloud_computing.aspx

[7] Guoli, Z., Wanjun, L. 2010. The applied research of cloud computing platform architecture in the E-Learning area. *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol.3, no., pp.356,359.

[8] Koding | Say goodbye to your localhost and code in the cloud. Retrieved February 2, 2015, from https://koding.com/

[9] Lifka, D., Foster, I., Mehringer, S., Parashar, M., Redfern, P., Stewart, C., & Tuecke, S. (2013). XSEDE cloud survey report. Technical report, National Science Foundation, USA.

[10] Lohmosavi, V., Nejad, A.F., Hosseini, E.M. 2013. E-learning ecosystem based on service-oriented cloud computing architecture. *Information and Knowledge Technology (IKT), 2013 5th Conference on* , vol., no., pp.24,29, 28-30.

[11] Sultan, N. 2010. Cloud computing for education: A new dawn? *International Journal of Information Management, 30*(2), 109-116.

[12] Thomas, P. Y. 2010. Cloud Computing: A Potential Paradigm for Practicing the Scholarship of Teaching and Learning. *The Electronic Library*, 29(2): 214–224.

[13] Web services Definition from PC Magazine Encyclopedia. Retrieved February 2, 2015, from http://www.pcmag.com/encyclopedia/term/54345/web-services

[14] Welcome to CDW•G. Retrieved February 2, 2015, from http://www.cdwg.com/

[15] Welcome to web.py! (web.py). Retrieved February 2, 2015, from http://webpy.org/