

# UNIFIED MODELING LANGUAGE TOOLS COLLABORATION FOR USE CASE, CLASS AND ACTIVITY DIAGRAM IMPLEMENTED WITH HTML 5 AND JAVASCRIPT FRAMEWORK

Aditya Kurniawan, Bina Bestina Harefa and Surya Sujarwo

School of Computer Science, Bina Nusantara University, Jakarta Barat, Indonesia

Received 2014-01-29; Revised 2014-02-03; Accepted 2014-03-23

---

## ABSTRACT

As a result of the development of web 3.0, web technologies now make it possible for each user to collaborate in performing a task. This technology allows us to draw UML diagrams online and collaborate on a software project. Unified Modeling Language is one of the architectural modeling software that is widely used by software developers. This research aims to develop a modeling tool UML diagrams are class diagrams, use case diagrams and activity diagrams based on pre-existing web using HTML 5 technology combine with JSON Service that allows the software developer to work on the same project UML and collaborate each other with good performance more faster than ordinary web.

**Keywords:** Unified Modelling Language, Use Case Diagram, Activity Diagram, Class Diagram, CASE Tool Collaboration, HTML 5

---

## 1. INTRODUCTION

### 1.1. Background

At this time a lot of research on the development of Case Aided Software Engineering (CASE) tools for chart a Unified Modeling Language (UML). UML CASE tool at this point is a desktop-based case tool that is only used to describe the individual diagrams. Beginning in 2000 CASE tools were developed using web technologies (Khaled *et al.*, 2002). Web-based UML CASE tool enables that CASE tool is used together and collaborate among software developers. This paper presents the development of CASE Tool on the use case diagram and class diagram by using web HTML technology and javascript on the application layer. Development of the case tool in this research is focused on class diagram and use case diagram.

Use case diagram is diagram that used to describe the features of the designed system and its relationship to system users. Use case diagram is also used as the initial description of a system that

correlates with other diagram such as activity diagram and sequence diagram.

Class diagram is a diagram that used to describe the architecture of the system with object-oriented methodologies. Class diagram is also a basic diagram to illustrate other diagrams like sequence diagrams, collaboration diagrams and object diagrams.

### 1.2. State of the Art

This research is the development of a web-based UML CASE tool by using the different technologies and processes. The process developed is the depiction of the use case diagram and class diagram in the UML project. The same UML project can be accessed by multiple software developers so that software developers can collaborate in designing use case diagram and class diagram.

The technology that is used in this research using HTML 5 combined with a javascript framework. In this research also used the JSON service layer approach that is used as a communication layer between the application layer to the database layer.

---

**Corresponding Author:** Aditya Kurniawan, School of Computer Science, Bina Nusantara University, Jakarta Barat, Indonesia

## 2. PREVIOUS RESEARCH

### 2.1. Related Works

Khaled *et al.* (2002) started a research interface for a web-based CASE tool in sequence diagram. CASE Tool by Khaled *et al.* (2002) is divided into two categories based on performance and functionality, the heavyweight and lightweight categories. CASE Tool heavyweight category is a desktop-based CASE tool that is isolated from computer networks such as Rational Software Architect that is used to describe the complex UML. CASE tool heavyweight category also has a lot of functionality that so much functionality is wasted. In addition this CASE tool can only be installed on each computer to use it. CASE Tool lightweight is a tool which has a sleek and lightweight functionality (Khaled *et al.*, 2002).

According to Mackay *et al.* (2003) the lack of CASE tool is not agile in creating UML diagram because of too much functionality to be used in a UML diagram project. The most fundamental difference between heavyweight and lightweight is the amount of functionality that is provided. Attribute key that identifies the CASE tool lightweight is essential or core functionality. Lightweight functionality including that does not restrict users to create UML diagram only on one desktop only. Therefore Khaled *et al.* (2002) use web technology. Web technology allows the CASE tool can be run in different desktop without having to install on a desktop computer. In addition web technology also enables applications run with better performance than heavyweight.

The architecture of the sequence diagram CASE tool of Khaled *et al.* (2002) contained in **Fig. 1** below.

This CASE tool uses a java server pages combine with java bean and java servlets technology and the database uses MySQL.

The use of UML diagrams are used in various fields. One of them is the Singh *et al.* (2010) research where this research utilizes UML to design a conceptual model of an event-based system with OOAD methodology. Sing *et al.* (2010) utilize the components of a class diagram to describe the events of meta models. Sing *et al.* (2010) also modify the class diagrams and use case diagrams diagrams in order to describe an event-based system and create rules specific depiction of the model diagram.

UML also develop for real-time system which known as UML-RT. This UML-RT use object diagram for draw structure model, sequence diagram to draw object interaction (Gao *et al.*, 2004).

Tilakratna *et al.* (2012) proposed an UML diagram that can constructed an ontology language with XML tags to represent the characteristic of real world system precisely at the analysis phase (Tilakaratna and Rajapakse, 2012).

### 2.2. Use Case Diagram

Use case diagram is used to obtain the behavior of the stakeholders involved with the system (Eichelberger, 2008). Use case diagram is visualized as high system functionality based on the viewpoint of the user (Eichelberger, 2008). Eichelberger (2008) explains that for drawing the graph, we can use the Sugiyama algorithm with the following steps:

- Layer Using as a depiction of the use of diagrams overlap
- Reduction of intersecting lines on the diagram
- Calculate the diagram's coordinate position

Hirschfeld *et al.* (2011) makes Use CasePy to make the development of software based on use case-centric which the use case is at the center of development. Use case is the main UML diagram in the depiction of system that can be used as a benchmark for other diagrams such as sequence diagram and activity diagram.

### 2.3. Class Diagram

Class diagram is the main diagram in object-oriented programming. The depiction of the class in this diagram consists of the class diagram along with the relationship among the class that consisting of aggregation, composition, directional, bidirectional and inheritance.

According Gutwenger *et al.* (2003) class diagram can be described as a class hierarchy diagram and non-hierarchy. The depiction of the class hierarchy diagram is usually used in commercial tools issued by the software engineering industry such as Rational Software Architect. **Figure 2** is an example of a class diagram depiction released by the industry as a hierarchically.

### 2.4. Activity Diagram

Activity diagram is currently used for the design specifications based on existing use case. Development of activity diagram can be used for the system to the field of economics or bio-informatics (Siebenhaller and Kaufmann, 2006). In addition to the research of Cui *et al.* (2009), an activity diagram is used to model the software development with aspect-oriented approach.

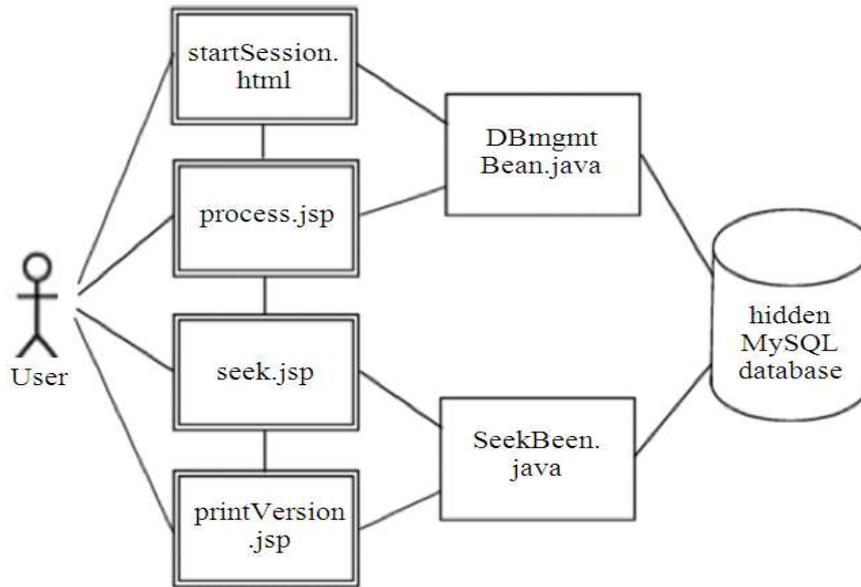


Fig. 1. Sequence diagram case tool (Khaled *et al.*, 2002)

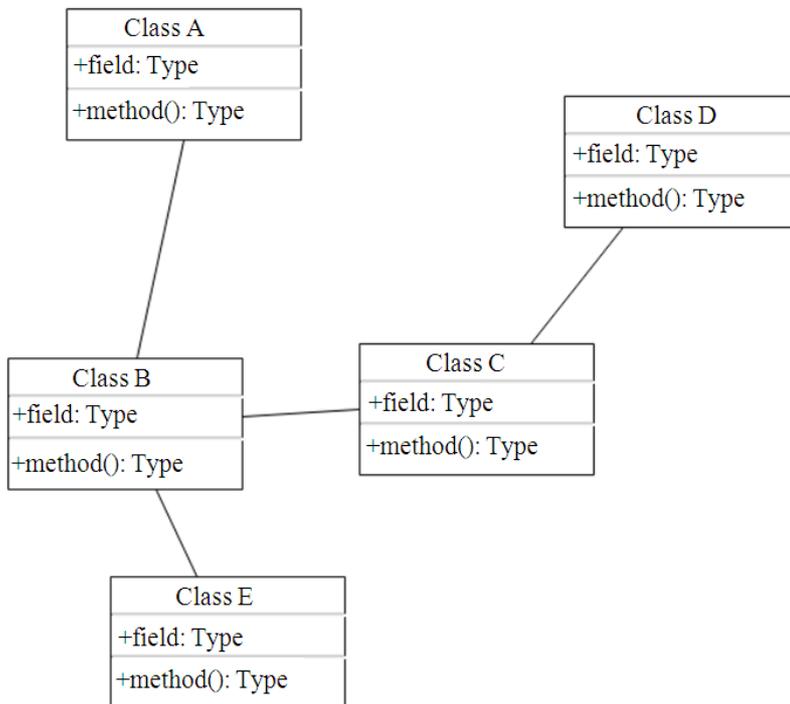


Fig. 2. Industrial layout in hierarchical style (Gutwenger *et al.*, 2003)

Standard layout and component are proposed by Ambler (2005). Important aspects in the depiction of an

activity diagram according to Ambler that is raised by Siebenhaller and Kaufmann (2006) is as follows:

- Minimize the number of lines intersecting
- Drawing lines orthogonally in the form of horizontal and vertical segments sequentially
- Reorganize the large diagram activity diagram into several smaller ones. In practice, an activity diagram has a maximum of 50 nodes and 80 lines

### 3. MATERIALS AND METHODS

#### 3.1. Method

This research uses the scrum methodology starts from the preparation of the framework of a dynamic architecture using JSON-based software services as the main engine is partially built. After that the next step to implement the algorithm on canvas depiction diagram HTML 5.

Code construction is done by making parts of the framework architecture into separate layers so as to allow the system to be much more developed.

This research also use MySQL as database because this research only focus on implement drawing UML diagram on canvas HTML 5 which could be collaborate with other user.

Trials this CASE tool system performance using YSlow and Page Speed to measure the speed of web load testing in the browse.

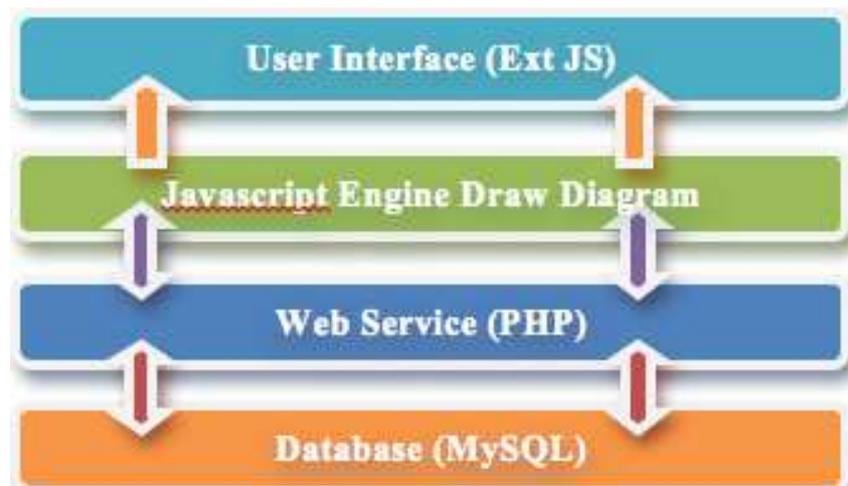
#### 3.2. Software Architecture of UML

The development of the CASE tool use software architecture approach by using JSON service as a

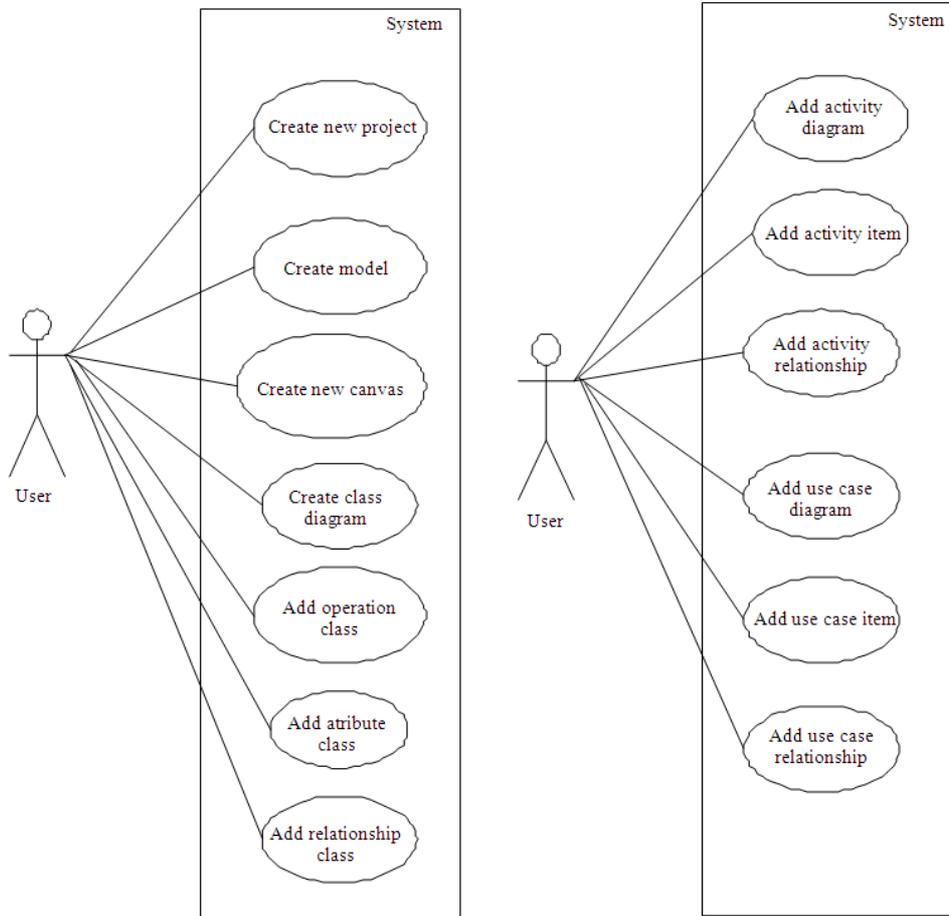
bridge layer between the database with HTML based and javascript case tool. In research of Sumaray and Makki (2012) concluded that JSON can be implemented to web service that is consumed in cross-platform and superior than the XML data format. JSON service will be created by using PHP technology that uses Code Igniter framework. Javascript is a client-side scripting web that be run on the most famous browsers. Javascript is famous because it can lighten the workload of the servers in the request process and server response. But javascript has a weakness that can be exploited easily because of its location on the client-side. Research of Guarnieri *et al.* (2011) makes ACTARUS which is website crawler that can exploit javascript's weaknesses such as the AJAX URL that is stored in the script. Therefore, in this research, javascript will be obfuscated i.e., javascript encryption through specific codes that javascript is not exploited easily.

ExtJs javascript framework is used to build the user interface of this tool case. ExtJs has the advantage of creating a web interface such as javascript based desktop. Additionally it also encodes javascript by using object-oriented concepts (Sencha, 2013). Ext Js components are menu component, treeview, region layout and image.

Software architecture in this research is illustrated in **Fig. 3**. The main engine of this case tool is in a PHP web service that is a bridge between the database and javascript engine diagram. The web service will generate JSON data that will be used by the javascript engine diagram to describe the diagram on user interface.



**Fig. 3.** Case tool software architecture



**Fig. 4.** Use case and case tool

This case tool is made in a collaborative tool, where among users can collaborate online to describe the diagram on the same project. **Figure 4** is a use case to illustrate the interaction of the user with the existing case tool features. In case this tool, the UML project consists of 3 models: class, activity and use case. One model can be consists of a lot of diagrams. In each diagram, there is diagram's item and relationship to generate diagram in HTML 5 canvas that is existing in the system.

**Figure 4** is a use case diagram for UML Collaboration. This diagram explain that user could make a canvas for drawing a diagram in one project and drag out diagram items and draw in canvas.

#### 4. RESULTS

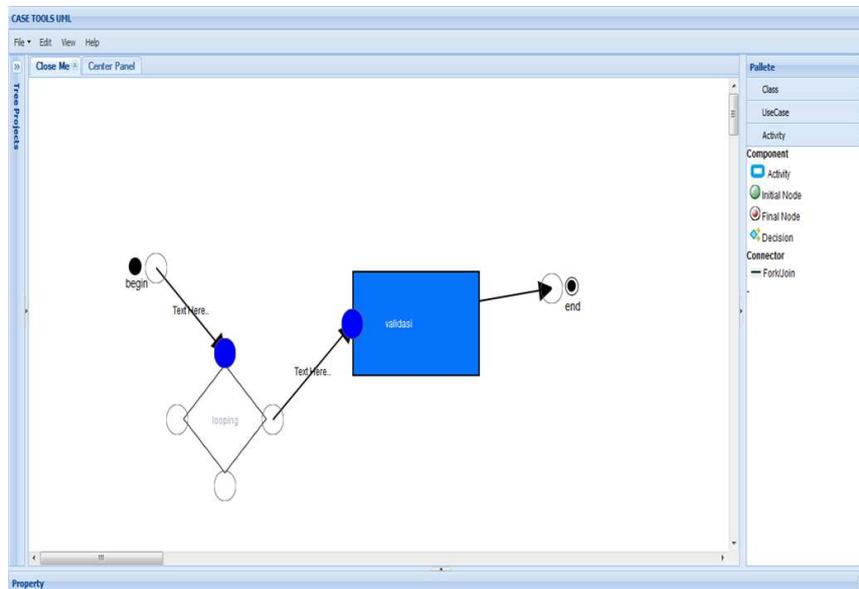
This research resulted the web-based case tool in the form of a desktop application by using javascript. **Figure**

**5** is a feature of depiction about activity diagram to connect the process nodes in the activity diagram.

On the layout in **Fig. 5**, on the top there is a menu and on the left there is a list of canvas diagram, model and project in the form of treeview. On the right there are the items that can be used on the canvas.

To evaluate the performance of this tool case, we use a YSlow tool that is released by Yahoo's and Page Speed tool that is released by Google as a measurement of performance. The results of tools measurement are listed in **Table 1**. From the experimental data showed that the experiment by using YSlow, this case tool gets 75 scores of 100 rated from existing web components by Souder (2007).

**Table 2** shows the results of measurements by using page speed. Page speed use the colors method i.e., red, green, yellow to measure the performance of each component of web measured.



**Fig. 5.** Diagram feature case tool

**Table 1.** Case tool performance with Yslow result

Grade	Item
F	Make fewer
A	Use a Content Delivery Network (CDN)
A	Avoid empty src or href
F	Add expires headers
F	Compress component with gzip
A	Put CSS at top
F	Put Javascript at bottom
A	Avoid CSS expression
n/A	Make Javascript and CSS External
A	Reduce DNS Lookup
F	Minify Javascript and CSS
A	Avoid URL Redirects
A	Remove duplicate javascript and CSS
F	Configure entity tags (Etags)
A	Make Ajax Cacheable
A	Use GET for AJAX request
A	Reduce DNS Lookup
A	Avoid HTTP 404 (Not Found) error
A	Reduce cookie size

The measurements by using page speed show the performance assessment of this case tool is 75 of 100. Web components are assessed on a different page speed are different with the assessment of web components contained in YSlow.

From both of performance measurements of these tools show that this web requires file compression technique and minify a better javascript.

**Table 2.** Case tool performance with page speed

Color Grade	Item
Red	Enable Compression
Red	Leverage browser caching
Yellow	Minify Javascript
Green	Minify CSS
Green	Defer Parsing of Javascript
Green	Minify HTML
Green	Optimize Images
Green	Specify image dimension
Green	Specify a character set
Green	Optimize the order of style and script
Green	Specify a vary: Accept-Encoding header
Green	Avoid CSS @Import
Green	Avoid bad request
Green	Combine images into CSS sprites
Green	Enable Keep-Alive
Green	Inline Small CSS
Green	Inline Small Javascript
Green	Make landing page redirects cacheable
Green	Minimize redirects
Green	Minimize request size
Green	Prefer asynchronous resource
Green	Put CSS in the document head
Green	Remove query strings from static resources
Green	Serve resource from a consistent URL
Green	Serve scale images
Green	Specify a cache validator

## 5. DISCUSSION

Case Tool from the results of this research has

been considerably improved from Khaled *et al.* (2002) and Mackay (2003) only describe UML diagrams online but can't collaborate. This system has implement drawing algorithm in canvas HTML 5 making this canvas possible to use on many user in one same project, so any developer could see and work collaborate their diagram project.

This system also have good score for this load web testing from Yslow (75) and PageSpeed (75). This score could elevate with fixing any grade web component below a in YSlow or any yellow and red color on PageSpeed.

This research is another approach for UML CASE tools which is using canvas HTML 5 technology and javascript combine with JSON Service.

## 6. CONCLUSION

Limitation of this research that its system architecture design based on ordinary web request and response so it less real time so user must need refresh page when design is change from another user

This research can enhance more with combining push data engine like XMPP and big data like Cassandra to make this collaboration diagram more real-time in the future of next study.

## 7. REFERENCES

- Ambler, S.W., 2005. The Elements of UML 2.0 Style. 1st Edn., Cambridge University Press, ISBN-10: 0521616786, pp: 188.
- Cui, Z., L. Wang, X. Li and D. Xu, 2009. Modeling and integrating aspects with UML activity diagrams. Proceedings of the ACM Symposium on Applied Computing, Mar. 08-12, ACM Press, Honolulu, HI, USA., pp: 430-437. DOI: 10.1145/1529282.1529377
- Eichelberger, H., 2008. Automatic Layout of UML use case diagrams. Proceedings of the 4th ACM Symposium on Software Visualization, Sept. 16-17, ACM Press, Munich, Germany, pp: 105-114. DOI: 10.1145/1409720.1409738
- Gao, Q., J. L. Brown and F.L. Capretz, 2004. Extending UML-RT for control system modeling. J. Comput. Sci., 338-347. DOI: 10.3844/ajassp.2004.338.347
- Guarnieri, S., M. Pistoia, O. Tripp, J. Dolby and S. Teilhet *et al.*, 2011. Saving the world wide web from vulnerable javascript. Proceedings of the International Symposium on Software Testing and Analysis, Jul. 17-21, ACM Press, Toronto, ON, Canada, pp: 177-187. DOI: 10.1145/2001420.2001442
- Gutwenger, C., M. Junger, K. Klein, J. Kupke and S. Leipert *et al.*, 2003. A new approach for visualizing UML class diagrams. Proceedings of the ACM Symposium on Software Visualization, Jun. 11-13, ACM Press, San Diego, CA, USA., pp: 179-188. DOI: 10.1145/774833.774859
- Hirschfeld, R., M. Perscheid and M. Haupt, 2011. Explicit use-case representation in object-oriented programming languages. Proceedings of the 7th Symposium on Dynamic Languages, (SDL '11), ACM Press, New York, USA., pp: 51-60. DOI: 10.1145/2047849.2047856
- Khaled, R., D. Mackay, R. Biddle, J. Noble and E. Tempero, 2002. A lightweight web-based case tool for sequence diagrams. Proceedings of the SIGCHI-NZ Symposium on Computer-Human Interaction, (CHI' 02), ACM Press, New York, USA., pp: 55-60. DOI: 10.1145/2181216.2181226
- Mackay, D., J. Noble and R. Biddle, 2003. A lightweight web-based case tool for UML class diagrams. Proceedings of 4th Australasian User Interface Conference User Interface, (CUI '03), ACM Press, Australian Computer Society, Inc., Darlinghurst, Australia, pp: 95-98.
- Sencha, 2013. The class system-Ext JS 4.2.1-sencha docs. Sencha.
- Siebenhaller, M. and M. Kaufmann, 2006. Drawing activity diagrams. Proceedings of the ACM Symposium on Software Visualization, ACM., New York, pp: 159-160.
- Singh, S.K., S. Shabarwal and J.P. Gupta, 2010. An event-based methodology to generate class diagrams and its empirical evaluation. J. Comput. Sci., 6: 1301-1325. DOI: 10.3844/jcssp.2010.1301.1325
- Souder, S., 2007. High Performance Web Sites: Essential Knowledge for Front-End Engineers. 1st Edn., O'Reilly Media, ISBN-10: 0596529309, pp: 170.
- Sumaray, A. and K.S. Makki, 2012. A comparison of data serialization formats for optimal efficiency on a mobile platform. Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, Feb. 20-22, ACM Press, Kuala Lumpur, Malaysia, pp: 48-48. DOI: 10.1145/2184751.2184810
- Tilakaratna, P. and J. Rajapakse, 2012. Ontological framework for object-oriented analysis and design. Am. J. Eng. Applied Sci., 5: 251-260. DOI: 10.3844/ajeassp.2012.251.260