Principles and Dynamics of Block-based Programming Approach

Siti Nor Hafizah Mohamad[1], Ahmed Patel[2], Yiqi Tew[3], Rodziah Latih[4], Qais Qassim[5]

[1, 2, 3, 4,5]School of Computer Science, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor Darul Ehsan, Malaysia

[2]Visiting Professor
Faculty of Computing Information Systems and Mathematics
Kingston University,
Penrhyn Road, Kingston upon Thames KT1 2EE, United Kingdom.

[1]fiza410 *at* gmail.com, [2]whinchat2010 *at* gmail.com, [3]yiqi01 *at* gmail.com [4]ggiemel *at* gmail.com, [5]qaisjanabi *at* gmail.com

***Abstract*- Block-based programming approach is based on the principles and dynamics of combining component-based programming approach with end-user programming paradigm which is purported to be more advanced and easier to use in practice. We perform a state of the art review of block-based programming by accessing the different forms of programming approaches and paradigms pertaining to it and establish the core fundamental principles, mechanics and dynamics with examples to illustrate the essence of this approach as well as any metrics to measure its performance. The important characteristics of block-based programming are gathered together and tabulated with clear definition of meaning and purpose. Correspondingly, we establish the criterion and metric by which to measure the effectiveness and efficiency of programming effort in developing software systems based on the block-based approach. In addition, we illustrate more on the block specification and identification to give brief understanding. Finally, we identify aspects of block-based programming which need further investigation, research and development.**

I. INTRODUCTION

Block-based programming approach is defined in the principles and dynamics based on combined approaches between component-based programming and end-user programming that are said to be more comprehensive and easy to be used in practice. In a world of computer programming, a block is declared as a section of a code combined together to consist of one or more declarations and statements [1]. Basically, a block is a program component being developed to be combined with other blocks to generate specific applications without the need of any programming skills. The block-based programming is an approach that integrates programming blocks for developing some applications in a short period of time and within estimated cost [2]. The main aim of this approach is to allow the application to be developed directly by the end-users.

The paper is comprised as follows: In Section 2, the background study on component-based programming and end-user programming is described in a simple manner with a review on their advantages and disadvantages to clearly judge their strengths and weaknesses. In Section 3, we present a state of the art literature review of block-based programming by comparing similar mashups composition tools such as Google App Inventor and IBM WebSphere sMash [3] which are using the same approaches as being studied in this paper. Section 4, shows the important characteristics of block-based programming which allow further understanding of the approach and is being explained by a clear definition of the meaning and purpose. We also identify the block by giving a real-example. Furthermore, we clarified the criterion and metric which was used to measure the effectiveness and efficiency of block-based programming of the software systems. Finally Section 5 offers a brief discussion and our conclusion for future work.

II. BACKGROUND

A. What is component-based programming?

Component-based programming approach is a rational addition of good code organizing principles where a component or software object is developed, encapsulated certain functionality or a set of functionalities [4]. This component will then be integrated into a larger piece of software. Essentially, this process is the concept that software components are written in such a manner that they provide functions common to many different systems. By using ideas from different hardware components, component-based software systems have to be flexible to accept new replacement of software component and functionality equivalent components where possible. Another significant characteristic of software component is reusability where they are to be designed, implemented and reused in many different programs.

Component-Based Development (CBD) is how to produce new software by merging *prefabricated components* with *new programs* that provide bonding between the components and their new functionality [5]. As shown in Table I, there are several key technical concepts of Component Based Software

Engineering (CBSE) such as component, interfaces, contract, component model and component framework [5].

TABLE I
KEY TECHNICAL CONCEPT OF CBSE

Key Concepts	Description
Component	A component is an obscure implementation of a contract of its provided interface, subject to the third party composition and conformant with component model. To implement such contract, the component may use services provided by other components which was called required services and are specified as a contract of an interface that is called the required interface. Component combines two perspective component as an implementation and component as an architectural abstraction.
Interfaces	Interfaces describe all of the properties of a component that probably guide to inter-component dependencies. Without providing any semantic information of the interface's designs and implementations, the interface proclaims a set of fields and a set of operation signatures.
Contract	A contract of an interface of component provides semantic information that identifies on how the interface can be used and allows them to define the dynamic behavior of the component on the interface. There are two senses of contract which are component contract and interaction contract. A component contract identifies a pattern of interaction rooted on that component. An interaction contract specifies a pattern of interaction among different roles and the reciprocal obligation of component that fill those roles.
Component Model	Specifies the design rules that must be obeyed by the component which improves the ability of the composition by encapsulating a variety of sources of interface mismatches through a common API.
Component Framework	Component framework provides services to sustain and impose a component model. The framework manages resources shared by components and provides the underlying mechanisms that allow interaction among components.

A Critical aspect of CBSE is the need for a system that will predictably demonstrate the quality attributes compulsory to them. The end user of commercial component technologies must rely upon off-the-shelf component models and framework to provide basis for achieving system quality attributes.

TABLE II
THE ADVANTAGES AND DISADVANTAGES OF COMPONENT-BASED APPROACH

Advantages	Disadvantages
• Eases work distribution • Eases maintainability • Parallel development • Extensibility	• Communication overhead • Development timing • Difficult integration due to the independency of the component

Although component technology vendors make claims concerning on how their products yield these attributes, the technical basis for these claim is being doubtful [5]. The main advantages and disadvantages of the component-based programming also can be referred to Table II [6].

B. What is End-user Programming?

Most people in a society will encounter experiences in a computer's world as "end-users" of package programs. Unfortunately the writers of these programs do not know the particulars of the work that the developers are trying to implement. Struggling to meet the needs of these diverse users, they expand their programs with hundreds of features most people never use. Life would be much simpler if each user could add the functions he or she wanted.

The programs must be designed to accept user-written components in appropriate places. There must be a method to store and manage them. Importantly, since most users do not have the time or awareness to study the tools and skills of a professional programmer, rational compromises are necessary. The clarity and overview of developed programming languages are trades for usability by a variety of metaphors and tricks. Metaphor helps to do the programming in much easier manner.

Since the appropriate metaphors have some of their capabilities and limitations which diverge commonly depending on the users and theirs purposes, there is no specific method of end-user programming. As a substitute, there is a variety of techniques, such as Programming by Demonstration, visual programming, and many domain-specific languages and formalisms. Ideally there is a smooth progression from simple but limited metaphors, to more complex and powerful techniques as the user-programmer advances.

TABLE III
THE ADVANTAGES AND DISADVANTAGES OF END-USER DEVELOPMENT

Advantages	Disadvantages
• Frees IS resources for higher priority projects • May help reduce the hidden backlog • Faster design/implementation cycle • More acceptable to users • Reduces communications problems between users and IS • Encourages innovation and creative solutions.	• Duplication of effort and waste of resources • Greatly increased costs • Loss of control over data • Loss of control of quality in both programs and data • Incompatibles prevent sharing • Can be used to circumvent control processes such as the steering committee • Generally produces narrow, inflexible systems with short lives.

End-User Development (EUD) is a study within the field of computer science and human-computer interaction which

explains the activities or techniques that allow amateur developers to produce or modify a software artifact. Early attempts in End-user development were concentrated in adding simple scripting programming languages to expand and familiarize oneself to an existing application, such as an office suite.

There are two basic reasons why EUD has being accepted:- one is that organizations are encountering delays on projects and by using EUD they can effectively reduce the time of completion of a project. The second is that software tools are more powerful and easier to use. Table III shows the main advantages and disadvantages of end-user development [7].

According to Nardi [9], three criteria are essential for end user programming to achieve a success level. The following three criteria are considered to be essential:

- A visual environment.
- A tack specific language.
- Support for collaborative practices.

III. State Of Art Literature Review

This review is to show the two new free mashup software tools which were released by Google called Google App Inventor for Android [10] and WebSphere sMash [11] which was released by IBM. These software tools were reviewed to show the similarity on using block-based approach in their application.

A. Google App Inventor for Android

Google App Inventor tool is intended to make it easy for people to build applications for its Android operating system. The tool enables people to drag and drop blocks of code which shown as graphic images and representing different Smartphone's capabilities and put them together, similar to snapping together Lego blocks. The result is an application on that person's Smartphone [10].

For example, one student made a program to inform a selected list of friends, with a short text message, where the user was every 15 minutes. The program was created by assembling three graphic code blocks together: one block displayed the phone's location sensor; another displayed a clock which he set for 15-minute intervals, and third linked to a simple database on a website, listing the selected friends.

In Figure 1, the screenshot which was taken from the website [10] shows the interview or the looks of the Google App Inventor which consist of several blocks. The 'Viewer' block which was in the center allows any element from the 'Palette' block to be *dragged and dropped* into it. The tool, specifically works only for phones which running Android operating systems. A sign-up with a Google Gmail is required. The tool is Web-based except for a small software download that automatically syncs the programs created on a personal computer, connected to the application inventor website, with an Android Smartphone. When making programs, the phone must be connected to a computer with USB link.

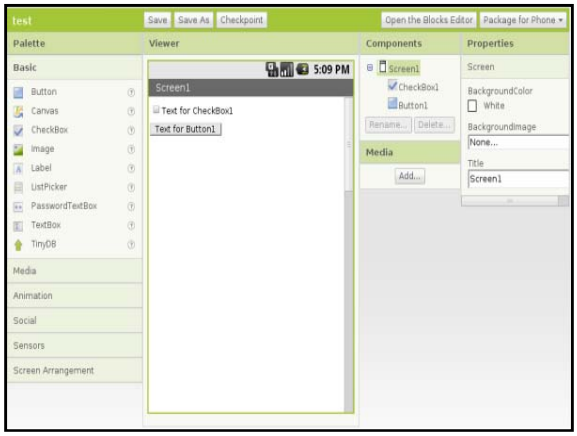

Fig. 1. The screenshot of the Google App Inventor

To use the App Inventor, a user does not need to be a developer. This is because instead of writing a code, the user visually designs the way the application looks and uses blocks to specify the application's behavior. The App Inventor team has created blocks for just about everything the user can do with an Android phone, as well as blocks for doing "programming-like" stuff such as blocks to store information, blocks for repeating actions, and blocks to perform actions under certain conditions. A block to talk to services such Twitter also had been made available for user.

B. IBM WebSphere sMash

IBM WebSphere sMash is an agile Web application platform that contributes in developing and running modern Web applications which introduces a simple environment for creating, assembling and running applications based on popular Web technologies.

IBM WebSphere sMash enables the following Web technologies [11]:

- A dynamic scripting runtime for Groovy and PHP.
- Java as a system programming language.
- Application programming interfaces optimized for producing Representational state Transfer (REST) services
- Rich Ajax Web user interfaces.
- Integration mashups and feeds.

TABLE IV
The Main Advantages Of WebSphere sMash

Advantages	Details
Speed	• Dynamic scripting languages. • Fewer lines of code. • Browser-based tooling.
Simplicity	• Leverage pre-existing content. • Use the web as your SOA platform. • Visual Assembly-style development.
Agility	• Small footprint. • Easily modify your applications. • Restart the server in seconds.

IBM WebSphere sMash software provides development and execution platform which based on the public incubator Project Zero [11]. Project Zero is an online development community that entitled the people who build and use this software. WebSphere sMash advances Smart Service Oriented Architecture's (SOA) properties and increase the alignment between Business and IT by allowing developers to rapidly deliver dynamic Web 2.0 based applications. Table IV shows the main advantages of IBM WebSphere sMash. The software's design is intended to increase the developer's productivity and efficiency through the support of dynamic scripting language [11].

Users can get several benefits from IBM WebSphere sMash software such:

- Improved developer productivity and efficiency by providing a runtime environment for Groovy and PHP scripts which allows developers to write applications and services leverage the assets of both Java and PHP community.
- Simplified deployment of applications which assembly the application to acts as the server and enable applications to connect to other resources through a Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS), Java Message Service (JMS) and Simple Mail Transfer Protocol (SMTP).
- Increased customer satisfaction and business productivity line by streamlining the end-users interaction with Web-based application which provides a visual editor for designing user interface where dragging and dropping of blocks is possible.
- Access to a growing online development community of consumers and the WebSphere sMash development team which allows the developers to interact directly with the development team, download the latest builds, submit bugs, access source code, read documentation and access to the latest browser-based tools for WebSphere sMash.

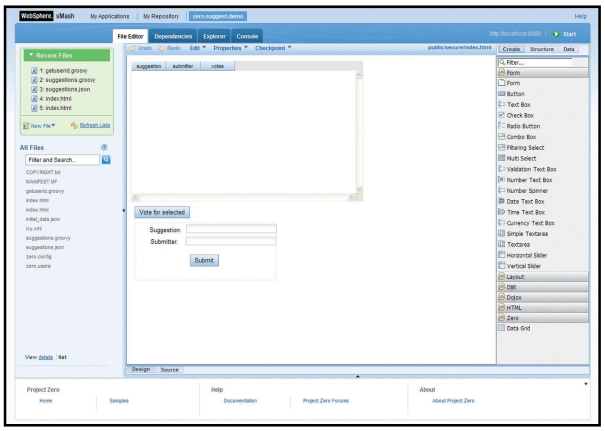

Fig. 2. The screenshot of the IBM WebSphere sMash

Figure 2 shows the screenshot of the web-based visual page editing tool in WebSphere sMash which applies the features of drag and drop design, automatic Hypertext Markup Language (HTML) source generation, widgets and Cascading Style Sheet (CSS) property editing and data binding. The screenshot was taken from the paper published by the IBM [11].

IV. Block-Based Programming (BBP) Approach

A. *Historical Background of BBP*

Block-based programming approach is a combined idea of component-based programming approach with end-user programming paradigm. The component-based programming attempts to generate tighter and better documented contracts so that the user can have some idea of what to expect from the component. Once the protocols between components are defined, this then allows the development of a component from different vendors to work together. Thus, one can buy off-the-shelf components from different vendors and plug it together into applications. In end-user programming, the user is allowed to develop web applications without knowing or writing the code. The concept of end-user programming is to provide and facilitate the users with controls such as pages, buttons, links, input fields, rules, and database records that the user needed to compose their applications.

The idea of combining these two programming is to reveal each of their advantages and eliminate each other's disadvantages. The component-based programming will effectively complete the job in a faster development mode. They also deployed in the real world. The programming is fairly safe, reliable, less code is needed to be written and also the plug-and-play systems are easy to modify. By combining the component-based programming and the end-user programming, the block-based programming will be able to satisfy the users with no programming skill to allow them to create their own application without even knowing any programming code. It's also allowed the block-based programming to be able to provide a powerful component from other vendor to plug it together as one application which later will satisfy the end-user.

Fundamentally, the term BBP is defined as an approach which integrating the programming blocks in a form to develop applications [11]. The main aim of BBP is to enable end-users to develop application directly. Several features for Block-Based Programming Approach have been identified as follows:

- Most of the blocks will be made available for a problem domain.
- Each block supports certain tasks.
- Users are allowed to customize blocks and to build applications adapting to their needs.
- Application software development can be performed by integrating those blocks.

The efficiency of any process approach is determined by the satisfaction of the consumer of its results. So, in terms of software development, the success is defined whether the software product can be produced on time, within budget, and the most important one, the continuous customer satisfaction. Even though, the continuity cannot be guaranteed, the high level of extensibility endorses it. In the Table V, shows the

advantages in terms of such attributes. Based on the initial characteristics of the component which was identified by Ian Sommerville [12], the desired characteristics of block-based programming as shown in Table VI.

TABLE V
THE ADVANTAGES OF BLOCK-BASED PROGRAMMING APPROACH

Advantages
• Existing applications are made possible to be reused. • High decrease of the associated application development costs. • Professional IT skills are not necessarily required. • Rapid application development is made possible. • There is no need to go for anyone's approval once a user wants to add an API to the internet. • Users' needs can be perfectly met by applications since the users are now able to incorporate content that they could not develop by themselves due to resource or time constraints.

TABLE VI
DESIRED CHARACTERISTICS OF BLOCK-BASED PROGRAMMING

Characteristics	Description
Standardized	Each block needs to conform to some standardized model which may include some attributes, variables, functions and classes.
Independent	Each block should be able to execute successfully without the need to depend on other blocks.
Composable	The block should provide external interactions and access to information about itself, such as attributes and method.
Deployable	The block has to be self contained and must be able to operate as a standalone entity.
Documented	The class or function library in the block has to be fully documented because a page of documentation is far more intelligent than a page of source code.
Testable	The block should conform to specific standards and to ensure the created block meets the user's needs.

In block-based approach, there are some criterion and metric to measure the effectiveness and efficiency of the programming effort in developing software system. Such criterions and metrics are:

- **Project management**: This explains how difficult or easy it is for the developer to manage the project in order to develop the system. In this case, it should be easy due to less work in programming because the block's service was made available by the third party.
- **Design quality**: The quality been measured in terms of effectiveness toward user's perspective. This shows how the design could help the user to understand the system easily.
- **Development speed**: These clarify that whether the system is able to execute within or over the timeline.
- **Testing**: This is to ensure the reliability of the system and to certify that all defined requirements are met. Such tests includes testing the performance of the application, testing each block, testing the integration of each blocks or finally testing the application in various environments.
- **Development cost**: This cost is basically considered on the complexity of the software system, the development environment and characteristic of the software system that is planned to be developed.
- **Maintainability**: This is based on how certain software systems manufactured were able to sustain correcting faults, to cope with new requirements, to make future maintenance easier or cope with a changing environment.
- **Reusability**: This indicates whether the modules and classes in the programming code could be used again to add new functionalities with slight or no modification which will helps in reducing implementation time, increase the likelihood that prior testing and use has eliminated bugs.
- **Satisfaction**: This includes the satisfaction from both of the developers such as the programmers and the end-users. The satisfaction was measured based on how the software system meets these users' requirement.

B. Block Identification

In a software application, especially while using the block-based programming approach, each of the blocks needs to be identified to show the individuality of each of them. Therefore, a block consists of one or more declarations and statements in terms of block's name, attributes, behaviors, variables, methods, classes, functions and interface which differentiate one block to another as shown in Figure 3.

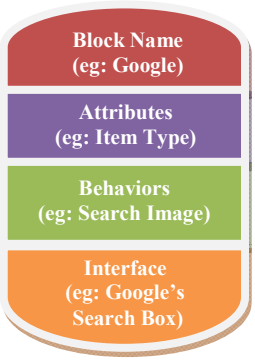

Fig. 3. Block Specification

In Figure 4, it shows the examples of block identification in an application by using declarations and statement. The example indicates the block services from Google Map and Youtube application services.

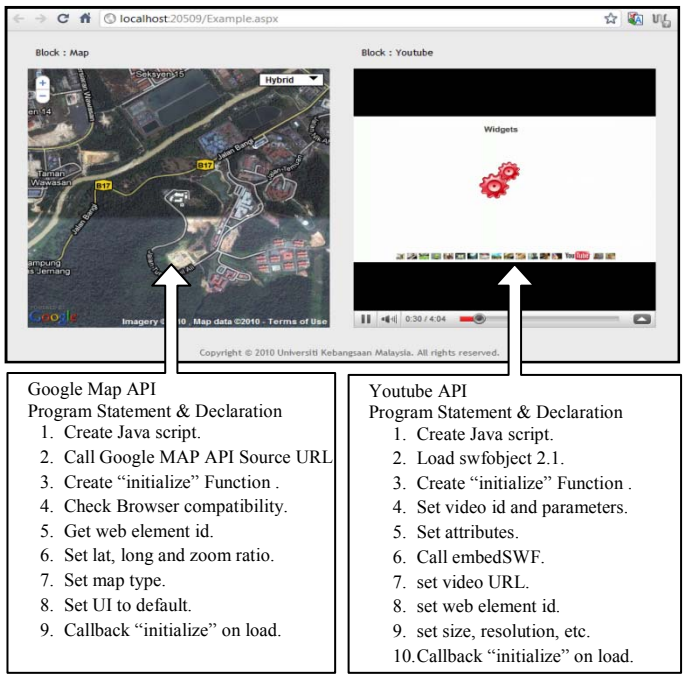

Fig. 4. Block Identification using statement and declaration

C. *The Proposed Model for Block-based Programming*

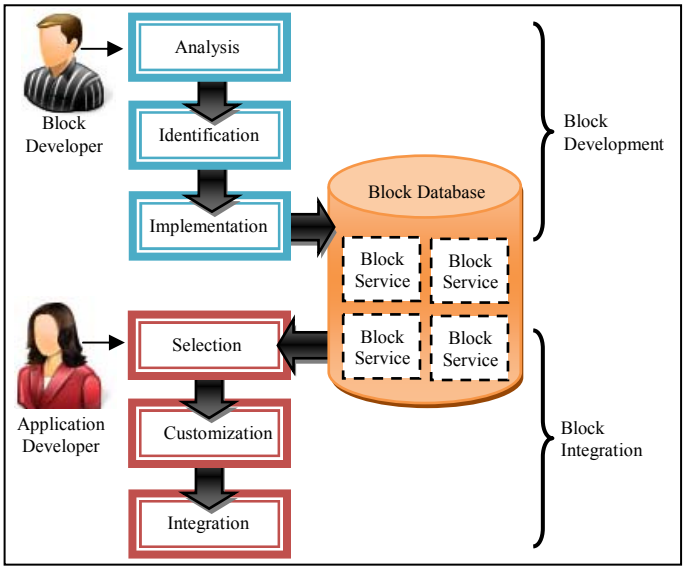

Fig. 5.The Proposed Model of Block-based Programming Approach

The proposed model for block-based programming form of two types of system developers, as illustrated in Figure 5, those will contribute to the block-based programming development approach. The first developer is the *Block Developer* (BD) or *Programmers*, who are responsible in identifying and implementing the blocks. The *Application Developers* (AD) or *end-users,* are responsible for selecting the suitable blocks from the list that was already developed by the BD. From the selection, the end-user will customize the block interface based on their needs and these selected block services will then be integrated as one application. The block database acts as storage for all blocks that have been developed by the BD and which will be used by the AD in order to create an application.

V. Conclusion And Discussion

In this paper, a CBP and EUP was been studied to understand fully the principles and dynamic of block-based programming approach. The block-based programming approach is a combined tool of the component-based programming approach with end-user programming paradigm. Also elaborated in the paper were the advantages and disadvantages of both component-based and end-user programming. The characteristics of block-based programming approach have been classified together with the metric involvement in the block-base to measure its the performance. Finally, the proposed model for block-based programming was introduced in order to develop the application. In future work, we will do further investigation, research and development on the framework architecture, and the programming involved in the block-based programming approach.

References

[1] Block (programming), Creative Commons Attribution-ShareAlike License. Wikipedia. http://en.wikipedia.org/wiki/Block_(programming), 2010. (Accessed 21-9-2010)

[2] A. Ismail, M. Djasmir, N. Omar, A. M. Zin. *Designing and Implementing Blocks for Developing Educational Software for Children with Learning Disabilities.* Universiti Kebangsaan Malaysia. 2009.

[3] Ahmed Patel, Liu Na, Rodziah Latih, Christopher Wills, Zarina Shukur and Rabia Mulla, 2010. A Study of Mashup as a Software Application Development Technique with Examples from an End-user Programming Perspective. Journal of Computer Science, Vol. 6, No. 11, pp. 1406-1415.

[4] A. Seyfi, A. Patel. *Briefly Introduced and Comparative Analysed: Agile SD, Component-Based SE, Aspect-Oriented SD and Mashups.* Data. International Symposium on International Technology (ITSim 2010). Volume 2, 2010, pp. 977-982.

[5] F. Bachman, L. Bass, C. Buhman, S. Comella-Dorda, F. Long, J. Robert, R. Seacord, K. Wallnau. *Technical Concepts of Component-Based Software Engineering.* CMU/SEI, Volume 2, 2000.

[6] M. Perez. *Component-based Development in Geographically Dispersed Teams*. MSE 17-652, 6 May 2002.

[7] H. Prahofer, D. Hurnaus, H. Mossenbock. *Building End-User Programming Systems Based on a Domain-SpecificLanguage*. Keba AG. Austria.

[8] Raymond S. Kulzick. *End-User Development: Advantages and Disadvantages.* Publication last modified: September 13, 2008, http://www.kulzick.com/endusrad.htm. (Accessed 26-9-2010)

[9] E. Kandogan, E. Haber, R. Barrett, A. Cypher, P. Maglio. *End-User Programming for Web-based System Administration.* UIST'05, October 23-27 2005, Seattle Washington.

[10] S. Lohr. *Google's Do-It-Yourself App Creation Software*. (2010), http://www.nytimes.com/2010/07/12/technology/12google.html?_r=1 (Accessed 18-10-2010)

[11] IBM Software Group. *IBM WebSphere sMash Version 1.1.* IBM Corporation. United State of America. 2008. http://www-01.ibm.com/software/webservers/smash/reqs/. (Accessed 30-10-2010)

[12] Ian Sommerville. 2006. *Software Engineering* Eighth Edition. Addison-Wesley, pp.443. ISBN: 7-111-19770-4.