



THE BLOCK STORE OF BLOCK-BASED PROGRAMMING APPROACH

MUSTAFA ALMATARY, MARINI ABU BAKAR AND ABDULLAH MOHD ZIN

Center for Software Technology and Management

Faculty of Information Science and Technology

Universiti Kebangsaan Malaysia

43600 Bangi, Selangor, Malaysia

E-mail: vim4mustafa@yahoo.com, marini@ftsm.ukm.my, amz@ftsm.ukm.my

ABSTRACT

Block-based Programming approach is an software development approach that makes programming task easier by enabling end user programmers to develop applications by integrating software blocks. In order for block based programming approach to be successful, there is a need for a large number of blocks to be developed in various application domains. These blocks must be developed based on a certain standard so that they can easily be combined with other blocks to create applications. These blocks must also be distributed and shared so that they are available for end users. This paper describes the concept and implementation of block store, which is a web based repository system that enables blocks to be managed and shared.

Keywords: *End-User Programming, Software Repository, Block-Based Software Development, Component-Based Software Development.*

1. INTRODUCTION

The common goal of software engineers over the last decade is the software reuse. In order for software to be reused, it must be structured into a collection of independent components. This approach of software development has a number of advantages: (i) It is easier to distribute components among various software engineers to allow parallel development; (ii) If components interrelations are clearly documented and kept to a minimum, software maintenance is easier since changes can be made locally without having unknown effects on the whole system; and (iii) It becomes easier to exchange components and incorporate new ones into a system.

Over the last decades, a number of approaches have been proposed to enable software to be reused, such as software libraries, CORBA, Java Beans, Microsoft COM and COTS components. Apart from that a number of programming systems have also been introduced such object-oriented programming, visual programming and natural programming to enable these components to be combined into applications.

Apart from that, there is also a movement towards end-user programming (EUP). One of the approaches of end user programming is where end-users are allowed to develop applications by customizing and combining components. In order

to support end user programming, there is a need for a new programming environment. A study by Myers (1986)[10] emphasized the important of improving the user interface to programming environment, so that we can move system design from “easy-to-use” to “easy-to-develop” and to make the process of software development more user-customizable. Researches that have been carried out show that there is a huge contribution of end-user programmers to software development (Lieberman et al. 2006 ; Spahn et al. 2008)[8][14]. Ko et al. (2004)[7] found that, as programming skills grow in demand and utility, the learnability of end-user programming systems is crucial. A survey by Beckwith & Burnett (2004)[3] indicates that EUP environments that support productizing, communication and sharing will be more valuable by end users. Reigeluth (1997) [11] states that we need to substitute standardization through customization, so computer-based learning software can create an interesting and simulating environment. A study carried out by Sutcliffe et al. (2003)[15] indicates that the EUP will be a mainstream competitor in the software development marketplace. Lieberman et al. (2006)[8] mentioned how at the political level EUP is important for full participation in the emerging Information Society.

One of the approaches for EUP is the Block Based Programming approach as discussed by Zin (2011)[16]. Block-based Programming approach is

a new approach that is proposed to make programming task become easier by enabling end user programmers to develop applications by integrating blocks. A block is actually a program that can be integrated with other blocks to build an application without going through the coding process. In the current implementation, a block is packaged as a JAR (Java Archive) file. JAR files provide a standard mechanism to compress and package a set of files for distribution to users.

In order for block based programming approach to be successful, there is a need for a large number of blocks to be developed in various application domains. These blocks must be developed by using a standard approach so that they can easily be combined with other blocks to create applications. These blocks must also be distributed and shared so that they are available for end users. In order to address these problems, this paper proposed and described the implementation of a block store.

2. RELATED WORKS

The concept of software repository is now being widely used by a software engineering community. The most popular one is the apps store, for example Apple App Store, Windows Store and Google Play. The apps store enables applications to be distributed and shared to a large number of users worldwide. It also provides standard guidelines to developers on how they could design, implement and share apps to broadest possible audience. The steps for developing and distributing apps differ from one store to another. For example in the Apple Apps Store the process goes through five steps: Enroll, Develop and Test the provision device, Create record, Submit apps and finally Release, as described in guide lines by Apple (2013)[2]. There are three processes in the Google Play: Design, Develop and Distribute (Android 2013)[1]. Windows store consists ten steps (Microsoft 2013)[9].

Another example of software repository is the one used by the open source development community such as the top 7 (Sourceforge, Google Code, GitHub, CodePlex, Eclipse, BitBucket, Java net). The aim of these open source repositories (Efytimes 2013) [4] is to bring different developers worldwide to contribute codes to some projects. It has been reported that the use of open source repositories has helped in the development of many software projects. However there are some issues that still need to be improved, such as, updating users with the latest version of the codes.

App Stores are basically targeting the apps users while Open Source Repositories are meant for professional software developers. Up to now there is no software repository that have been develop to support end user software development..

3. BLOCK-BASED SOFTWARE DEVELOPMENT

The block based development life cycle is shown in Figure 1

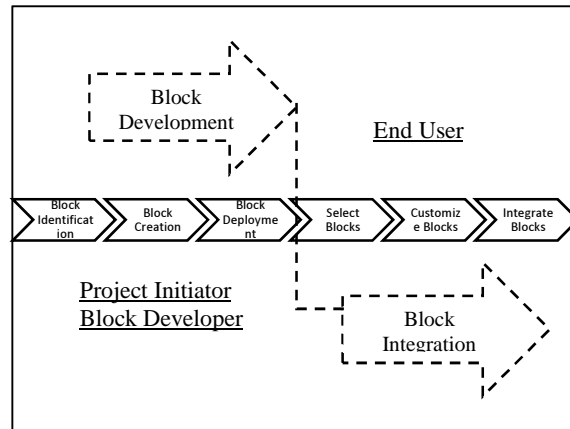


Figure 1: Blocks Based Development Life Cycle

Block Identification: The process of block identification consists of the following activities:

- ✓ Domain Analysis
- ✓ Domain Selection
- ✓ Identifying System Boundary
- ✓ Project identification
- ✓ Subsystem Analysis
- ✓ Block Identification and Composition.

Block Creation: Once blocks have been identified, they can be developed by writing program codes. In the current implementation, blocks are developed by using Java.

Block Deployment: The next process is for blocks to be deployed. Before these blocks can be deployed, they need to be certified to ensure that they are correct and safe.

Block Integration: End users can select blocks, customize and integrate them to build applications.

4. THE CONCEPTUAL FRAMEWORK OF THE BLOCK STORE

A block store is a web based system that enables blocks to be managed and shared. The conceptual framework for the block store is shown in Figure 2.

In the proposed framework, the block store consists of three components: people or players, methodology and tools.

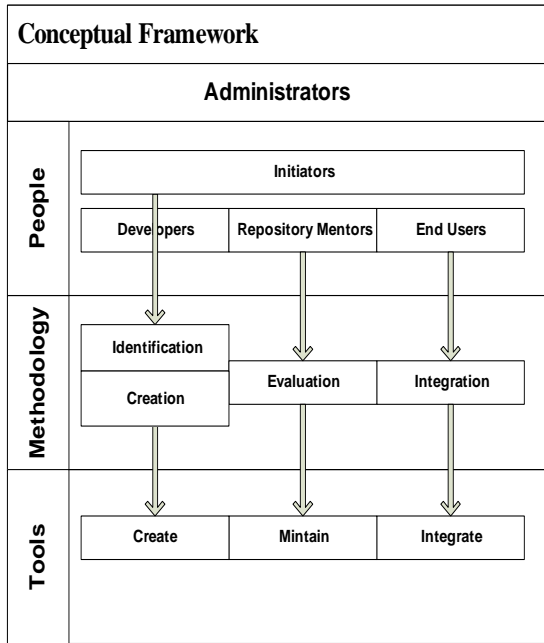


Figure 2: The Conceptual Framework for the Block Store

4.1 PEOPLE OR PLAYERS

From Figure 2, we can identify five main players involve in block-based software development: project initiators, block developers and end-user programmers, system administrator and visitors as shown in Figure 3.

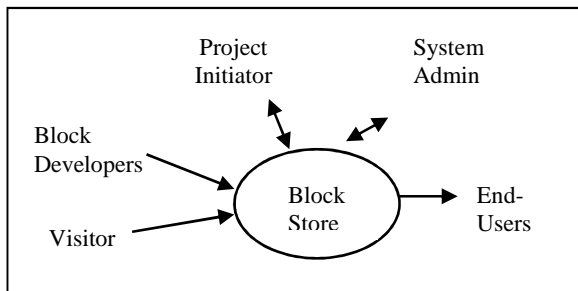


Figure 3: Players Involve in Block Store

Administrators: The responsibility of administrator is to manage users accounts (activate/inactivate or changing an account type into an initiator as per request) and creation of domains/subdomains (approvals/suspensions). The administrator also manages users profiles, authentication information and handles communication with all users through inbox messaging.

Initiators: The responsibility of the initiators is to manage their own domains and subdomains (create domain/subdomains, identify the required blocks and managing blocks submitted by developers). Initiators also manage their profiles, authentication information and communication with developers.

Developers: Developers are responsible for developing the required blocks in any domain of interest.

End Users: End Users browse the block store to obtain required blocks. These blocks can then be customized and then integrated to form an application. End users also manage their profiles and authentication information.

Visitors: Visitors are any users who can browse the block store. They do not have the permission to contribute blocks or to download blocks from the block store

These players can be organized similar to the concept of the onion OSS (Open Source Software) team structure as shown in Figure 4. The administrator has the highest level of authorization, followed by the initiators, developers, end users and finally the visitors.

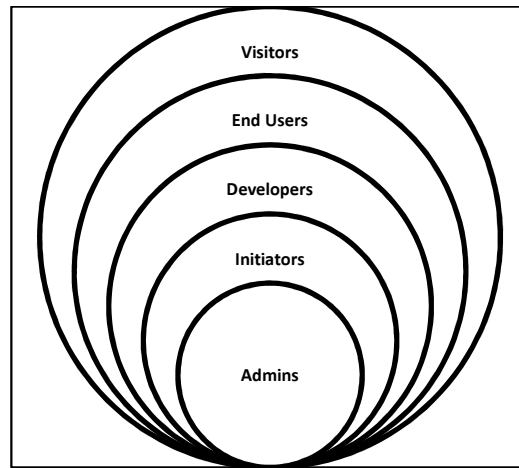


Figure 4: The Block Store Team Structure

Roles of each of these players can be shown by using a use case diagram as shown in Figure 5 at the end of this paper.

4.2 METHODOLOGY

Blocks development consists of a two stages as shown in Figure 6. The stages are the blocks identification and block creation. Block identification consists of the following activities:

- ✓ Domain analysis
- ✓ Subdomain Selection
- ✓ Block Identification

Block creation consists of the following activities:

- ✓ Identify Use Cases
- ✓ Identify Classes
- ✓ Identify Relationship
- ✓ Block Refinement

In order for these two activities to be carried out properly, there is a need for the right methodology to be in place. For block identification there is a need for Block Identification Methodology while for block creation there is a need for Block Creation Methodology.

Before a block can be put into the block store there is a need for the block to be properly evaluated and certified. Thus there is also a need for Block Evaluation Methodology.

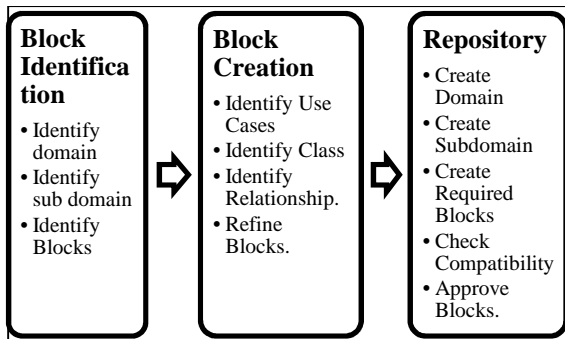


Figure 6: Stages of block development

4.3 TOOLS

The third component of the conceptual framework is the tools. According to Zin (2011) [16] there are a number of tools required to support the Block-Based Programming Approach. Two of the tools are the block creation and block integration tools. The block creation tool can be used to develop blocks by developers. An example of a block creation tool has been implemented by Djasmir et al (2012) [4]. The second tool is the block integration tool. We have identified two approaches for blocks integration. The first one is based on drag and drop concept that has been implemented Sarif et al. (2011) [12] and the other one is based on story telling approach that has been implemented by Hariri et al (2011)[6].

Tools for block certification need to be properly developed. This tool can be implemented by using the concept of program understanding (Sani,2009) [13].

5 DESIGN AND IMPLEMENTATION

The block store has been implemented as a web based system by using DHTML, where PHP and java scripting languages are used for managing the dynamic behaviors of the system. The MySQL data base is designed to store and manages the metadata and information of the block store. Furthermore, the evaluation mechanism handled by using the Java Servlets to support an automatic evaluation. Thus, the following sections explain the main functionalities of the system.

5.1 Operations

There are seven operations supported by the block store are:

- ✓ **Manage Account:** for the security and stability of the repository, the user's accounts need to be managed. Where a user type can be changed from developer to initiator depend on contribution and consistency of the user. Also, improper activities by users need to be managed where account can be disabled if necessary. Furthermore, each user has to manage his profile account details.
- ✓ **Manage Domain:** to ensure uniform process include a wider zone of development activities the domains needed to be instantiated by initiators and managed by them, therefore, the management of the domain activities such as (creation, edit, approval and suspended, etc.) is required.
- ✓ **Manage Sub-Domains:** for the categorization of the blocks development, the sub domain management operation also implemented where activities such as (creation, edit, approval and suspended, etc.) is designed.
- ✓ **Manage Blocks:** the blocks required is analysed by initiators for developers then according to that the developers create blocks and submit them into repository to be reused by end users. Therefore, the activity of identification and distribution required deferent authentication and the

- approval of distributed blocks required a proper management mechanism.
- ✓ **Authentication:** each user has the authority to manage his blocks, subdomains, domains according to the user's type and scope of development. Furthermore, the team structure as shown in figure 4 organizes those capabilities.
- ✓ **Manage Profile:** the user profile consists the registration information; therefore, to manage the login information and personal details this operation is created.
- ✓ **Messaging:** the user's communication provide in this section in order to encourage different users to communicate between one another.

5.2 Users

All users must register before they can enter the block store. There are five different types of users. Operations supported for each type of users is shown in table 1.

Table 1: the Block Store Users and Operations

Users Operations	Admin	Initiator	Developer	End User	Visitor
Manage Account	Enable, Disable, Change type, Communicate	Login, Register	Login, Register	Login, Register	Login
Manage Domains	Approve, Suspend	Create, Edit	Explore domains	Explore domains	View domains
Manage Subdomains	Approve, Suspend	Create, Edit	Explore subdomains	Explore subdomains	View subdomains
Manage Blocks	Brows	Analyse requirement, Approve, Suspend	Develop, Submit	Download, Customize, Integrate	View
Authentication	Enable	Change Login	Change Login	Change Login	Change Login
Manage Profile	Edit	Edit	Edit	Edit	submit
Messaging	Send, Delete	Send, Delete	Send, Delete	Send, Delete	Send, Delete

6. A CASE STUDY

We illustrate the use of the block store through a case study.

As shown in Table 1 the main blocks store operations performed by the different users and these operations performed in different levels of authentications. Before the operation can be performed, a user must be registered and has a login account. Figure 7 shows the login and registration form. During registration a user can choose either to be an initiator, a developer or an end user. The decision to approve a user registration is done by the block store administrator.

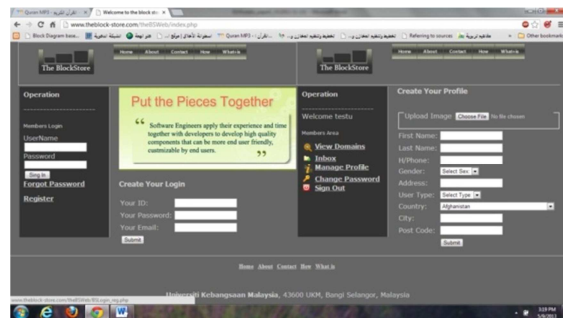


Figure 7: Create Login and Membership profile.

Suppose that "User1" applies to be an initiator, "User2" applies to be a developer and "User3" applies to be an end-user.

After the application is approved, "User1" will be able to manage domains, subdomains and submit the required blocks needed for the domain. Figure 8 shows how an initiator can instantiate the domain, subdomain and blocks required. Introduction of new domain requires approval from the Administrator. This will ensure that at one particular time only one initiator is responsible for one domain.

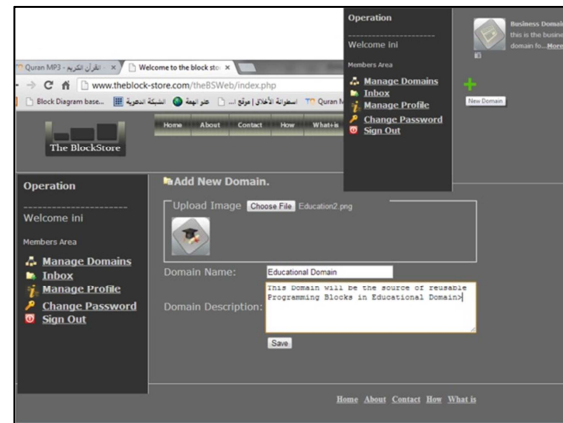


Figure 8: Creating new domain by initiator

Once the domain and sub-domains have been created, the initiator can now carry out the domain analysis to identify blocks required for that particular domain or sub-domain. All blocks required must be properly specified by the initiator. An example of block specification is given in Figure 9.

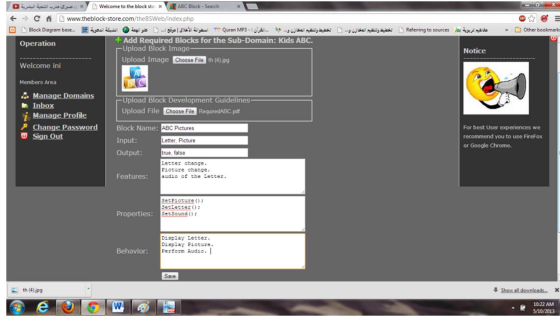


Figure 9: Block Specification

Once blocks have been specified, the information will be available for developers as shown in Figure 10. Developers can participate by developing the required blocks. The developed blocks can be submitted to the block store. However, these blocks will remain invisible for end users until they are approved by the initiator.

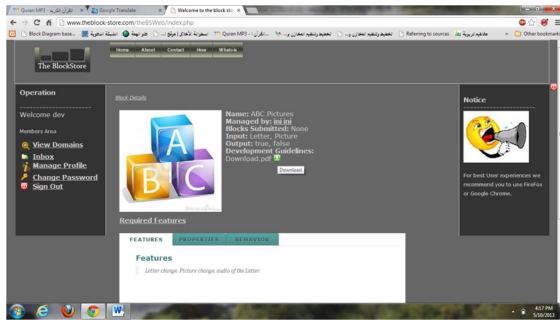


Figure 10: Block Specification as seen as Developers

Approved blocks will be available to be selected by end-users as shown in Figure 11. Some of these blocks are available for free while some of them may need to be purchased. Selected blocks can now be used by end-users for application development.

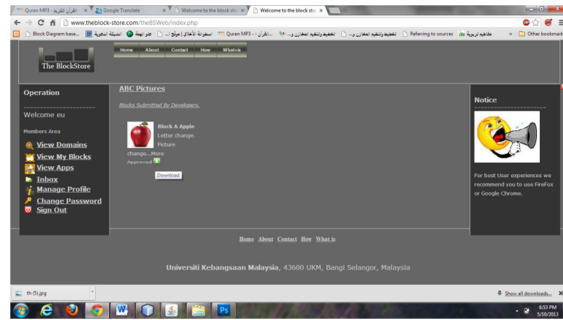


Figure 11: List of available blocks

7 CONCLUSIONS

Block-Based Software Development approach offers a software development environment that enables end users to make use of the available blocks to develop applications. This paper describes the design and implementation of the block store. The main purpose of the block store is to enable software developers to share and distribute blocks and to allow end users to browse and select required blocks. The availability of the block store enables end users to develop applications that satisfy their requirements. The block store also standardized the way blocks are identified, created and certified by enforcing the use of the right methodologies and tools.

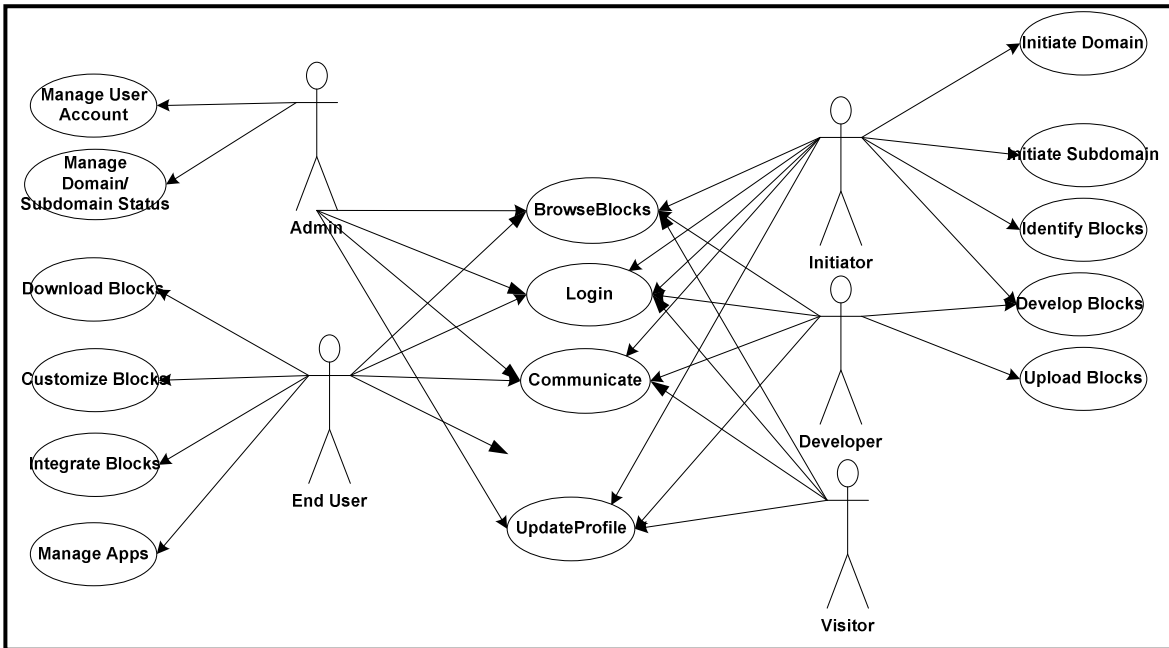


Figure 5: the Block Store Team Structure Use Case diagram

REFERENCES:

- [1]. Android, G. 2013. Publishing Overview. http://developer.android.com/tools/publishing/publishing_overview.html [17th Sep 2013].
- [2]. Apple. 2013. About App Distribution. <https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html> [17 th Sep 2013].
- [3]. Beckwith, L. & Burnett, M. 2004. Gender: An Important Factor in End-User Programming Environments? , 2004 IEEE Symposium on Visual Languages and Human Centric Computing, Rome, Italy, 26-29th September, pp. 107-114.
- [4]. Djasmir, M., Idris, S., Bakar, M. A. and Zin, A.M. 2012. An Integrated Development Environment for Blocks Creation. Asian Journal of Information Technology. Volume: 11 Issue: 6 Page No. 194-200
- [5]. Efytimes. Aug 27, 2013. Top 7 Open Source Project Repositories. <http://www.efytimes.com/e1/fullnews.asp?edid=114427> [Sep 17,2013].
- [6]. Hariri, H.M. , Bakar, M. A. and Zin, A. M. 2011. Story Telling Approach for Integrating SoftwareBlocks. Proceeding of the International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 17-19 July.
- [7]. Ko, A. J., Myers, B. A. & Aung, H. H. 2004. Six Learning Barriers in End-User Programming Systems. Proceeding of the IEEE Symposium on Visual Languages and Human Centric Computing, Rome, Italy, 26-29th September, pp. 199-206.
- [8]. Lieberman, H., Paternò, F., Klann, M. & Wulf, V. 2006. End-User Development: An Emerging Paradigm. In. End User Development, pp. 1-8. Springer.
- [9]. Microsoft. 2013. Overview of Publishing an App to the Windows Store. <http://msdn.microsoft.com/en-us/library/windows/apps/jj657972.aspx> [17th Sep 2013].
- [10]. Myers, B. A. 1986. Visual Programming, Programming by Example, and Program Visualization: A Taxonomy. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Boston, Massachusetts, USA,
- [11]. Reigeluth, C. M. 1997. Educational Standards: To Standardize or to Customize Learning. Phi Delta Kappan Vol. 78 Issue 3, pp. 202-206.
- [12]. Sarif, S. N., Idris, S. and Zin, A. M. 2011. The Design of Blocks Integration Tool to Support End-User Programming. Proceeding of the International Conference on Electrical



- Engineering and Informatics (ICEEI), Bandung, Indonesia, 17-19 July.
- [13]. Sani, N.F.M., Zin, A.M. and Idris, S. 2009. Implementation of CONCEIVER++: An Object Oriented Program Understanding System. *Journal of Computer Science*. 5 (12): pp. 1012-1022.
- [14]. Spahn, M., Dorner, C. & Wulf, V. 2008. End User Development: Approaches Towards a Flexible Software Design. *Proceedings of the European Conference on Information System, Ireland, 9-11th June*.
- [15]. Sutcliffe, A., Lee, D. & Mehandjiev, N. 2003. Contributions, Costs and Prospects for End-User Development. *Proceedings of the Tenth International Conference on Human-Computer Interaction, Crete, Greece, 22-27 June*, pp. 622-626.
- [16]. Zin, A. M. 2011. Block-Based Approach for End-User Software Development. *Asian Journal of Information Technology* 10(6): 249-258.