

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221013565>

Block-based programming approach: Challenges and benefits

Conference Paper · July 2011

DOI: 10.1109/ICEEI.2011.6021507 · Source: DBLP

CITATIONS

25

READS

4,676

6 authors, including:



Ahmed Patel

Universal Universities, Earth

248 PUBLICATIONS 3,490 CITATIONS

[SEE PROFILE](#)



Rodziah Latih

National University of Malaysia

11 PUBLICATIONS 350 CITATIONS

[SEE PROFILE](#)



Qais Saif Qassim

University of Technology and Applied Science

42 PUBLICATIONS 680 CITATIONS

[SEE PROFILE](#)



Na Liu

National University of Malaysia

2 PUBLICATIONS 57 CITATIONS

[SEE PROFILE](#)

Block-based Programming Approach: Challenges and Benefits

Siti Nor Hafizah Mohamad¹, Ahmed Patel², Rodziah Latih³, Qais Qassim⁴, Liu Na⁵, Yiqi Tew⁶

*School of Computer Science, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor Darul Ehsan, Malaysia*

¹fiza410@gmail.com, ²apatel@ftsm.ukm.my, ³rodziah@ftsm.ukm.my, ⁴qaisjanabi@gmail.com,
⁵hinadalu@hotmail.com, ⁶yiqi01@gmail.com

Abstract— Block-based programming approach is a combination of component-based programming approach with end-user programming paradigm that considered being more advanced and easier to use in practice. The aim of this paper is to discuss the benefits that block-based programming approach will be contributed to the community and the challenges that have to be solved in this new environment. In this paper, we introduce several previous good examples of applications that have been developed by using block-based programming approach. The examples are evaluated and compared in term of their development speed, design quality, usability, reusability, block customization, user acceptance, application platform and range of users. Finally, based on the evaluation and comparison, important challenges and benefits are specified and discussed.

Keywords— Benefits, Block-based Programming Approach, Challenges.

I. INTRODUCTION

Generally, block-based programming approach is 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 [1]. In addition, a block-based programming is an approach that allows integration several programming blocks for developing some applications in a short period of time, within estimated cost and without any programming skills needed [2]. Briefly, the component-based programming is a rational addition of good code organizing principles where a component or software object is developed, encapsulates certain functionality or a set of functionalities [3]. While end-user programming is an approach which explains the activities or techniques that allow amateur developers to produce or modify a software artifact [1]. The main aim of this paper is to discuss the benefits that block-based programming approach will contribute to the community and the challenges that can be solved with block-based environment.

The paper is comprised as follows: In Section II, the background study on block-based programming approach is described in a simple manner with a review of block application structure. In Section III, we identified the

important characteristics of block-based programming approach which allows further understanding of the approach. Section IV, shows the examples which have been evaluated and compared. Section V proposes a conclusion and brief discussion

II. BACKGROUND

A. Block-based Programming Approach

The main idea of combining component-based programming approach with the end-user programming approach is to preserve each approach's advantages and eliminate each other's disadvantages. 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 components. Once the protocols between components are defined, this then allows the development of a component from different vendor systems to interwork. 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 their code. The concept of end-user programming is to provide and facilitate the users with graphical controls such as pages, buttons, links, input fields, rules, and database records that the user needed to compose their applications [1].

Basically, the term block-based programming is defined as an approach which integrates the programming blocks into a lego-type form to develop applications [4]. In block-based approach, most of the blocks will be made available for a problem domain and each of the blocks support certain task. Then, the users are allowed to customize blocks and build applications adapting to meet their needs. The customization among the blocks will be performed by integrating those blocks.

There are several advantages of the block-based programming approach that can be briefly identified [1]:

- Developed applications can be reused.
- Minimizing and reducing the cost of application development to an acceptable level.

- Not necessarily required the professional IT skills.
- Rapid application development is made possible.
- No need for approval once a user wants to add an API to the application.
- Users' needs can be met perfectly.

B. Characteristics of Block-based Programming Approach

The efficiency of any process approach is determined by the satisfaction from the customer in term of the results. So, in terms of software development, the success is defined whether the software productions can be produced on time, within budget, and the most important one, the continuous customer satisfaction.

Fig. 1 shows the main characteristics of the block-based programming approach. The characteristics carry out the meaning as follows:

- *Standardized*: The blocks must conform to some standardized model.
- *Independent*: The blocks must be able to execute successfully without the existence of other blocks.
- *Composable*: The blocks should provide transparent internal action with interfaces to external interactions and access of adjacent blocks.
- *Deployable*: The blocks must be standalone entity.
- *Documented*: The blocks must have a fully documented library for the function or class.
- *Testable*: The blocks must ensure that it meet the user's needs and requirements.

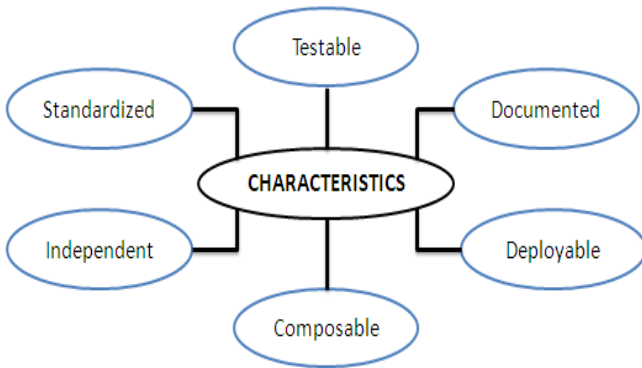


Fig. 1 Characteristics of Block-based Programming Approach [5]

C. The Structure of Block-based Programming Approach

As shown in Fig. 2, the structure consists of several steps that need considering while implementing an application by using block-based programming approach. The structure is divided into two major parts which are *block development* and *block unification*. The first three steps of block development consist of *analysis*, *identification* and *implementation*. While the steps for block unification include all the last three steps which consist of *block selection*, *block customization* and *block integration*. The block storage indicates as the database for storing the information for the available blocks.

In order for the application to be successfully implemented, it needs to be developed by two types of developers. The first developer is a block developer or also known as software programmer, which is responsible for the step in the block development part. The second developer is the application developer or globally known as end-users, and they are responsible for the step involves in the block integration [1].

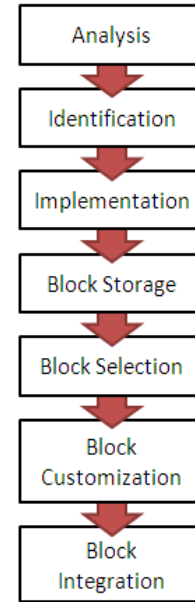


Fig. 2 Structure of Block-based Programming Approach

D. The Sketch Design of the Block-based Programming Approach

As shown in Fig. 3 and Fig.4, the design of the block-based approach consists of several features such as *block space*, *add block* button and also the *list of blocks*. When the *add block* button is click, the system will bring the application to the *list of blocks* which illustrated in Fig. 4. The *list of blocks* consists of the blocks that are made available by the block developer. After the block is selected, they will be appeared and published directly at the *block space*. In the *block space* area, the *block customization* and *block integration* will take place. This is where the end-user can customize the block based on their special needs. At the end, the application will only display and publish the application that the user intend to see and know.

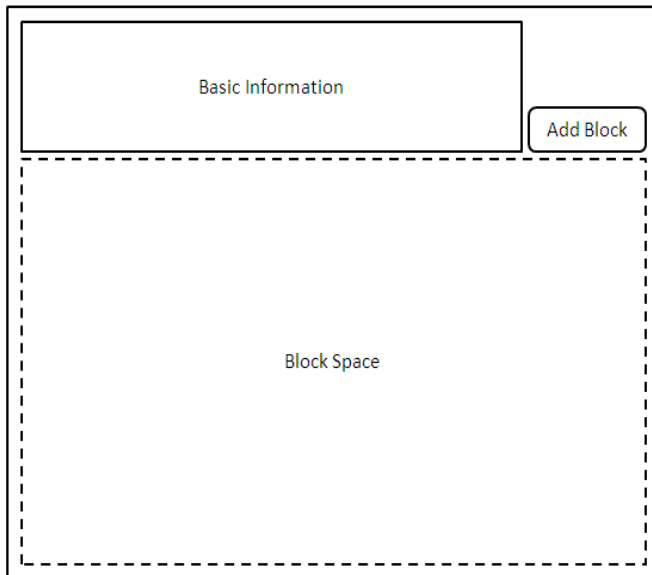


Fig. 3 Sketch design of the Block-based Programming Approach

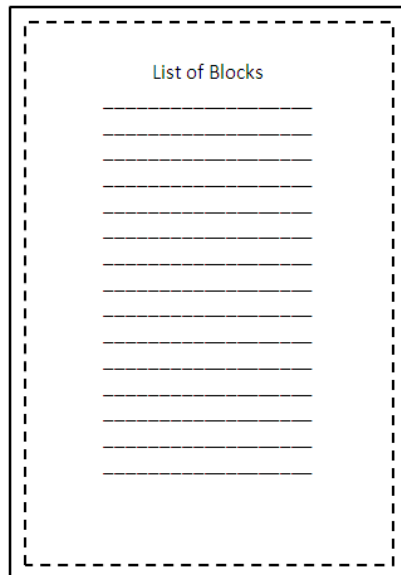


Fig. 4 Sketch design for the available blocks

III. THE BLOCK-BASED PROGRAMMING APPROACH OUTCOMES

This paper discusses three examples; the first example is an application which was developed by Google. The application named as Google Earth and this application allows user to travel the world through a virtual globe and view satellite imagery, maps, terrain, and 3D buildings [6]. In Fig. 5, it shows the interface of the Google Earth. The red-dotted box on the right of the interface indicates the block space. In addition, user needs to install a plug-in to be able to use the application.

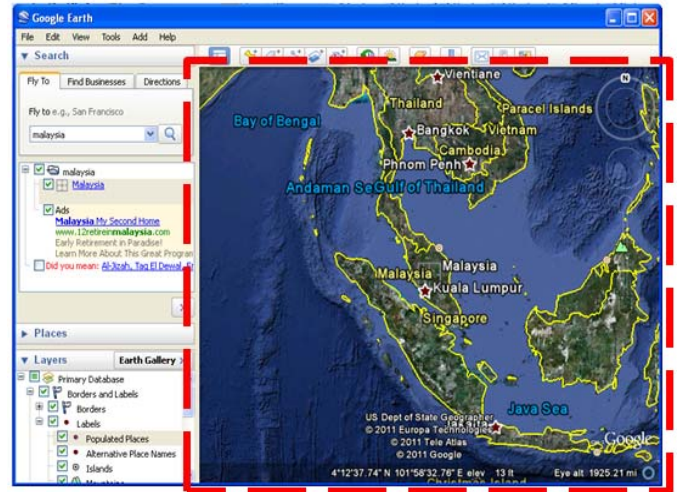


Fig. 5 Interface of the Google Earth

The second example of the application is an application from TweetDeck. TweetDeck acts as user's personal real-time browser for staying in touch with what's happening now, connecting the user with their contacts across Twitter, Facebook, MySpace, LinkedIn and many more [7]. Fig. 6 shows the interface of the TweetDeck. The red-dotted box on the bottom of the interface specified the block space for the application. In order for the block space in Fig. 6 to be filled with blocks, user needs to add block from the available listed block which is shown in Fig. 7. In this example, TweetDeck named the block as Column.

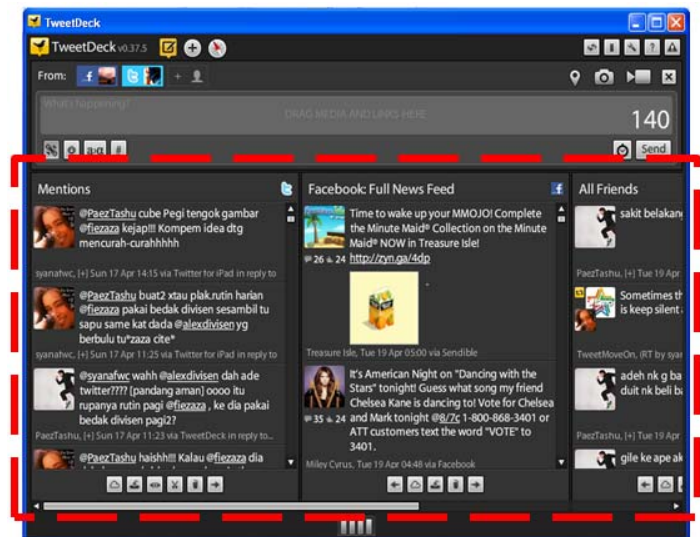


Fig. 6 Interface of the TweetDeck

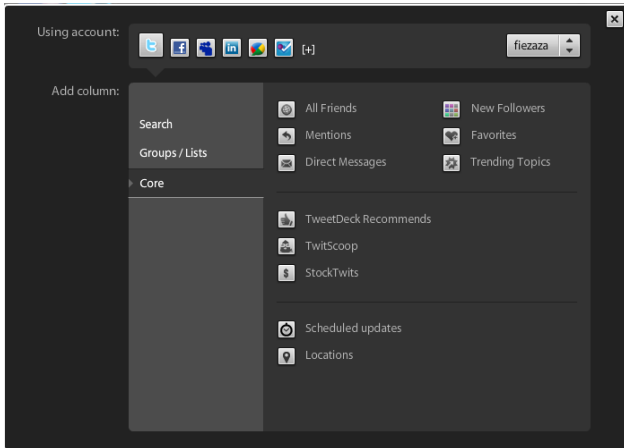


Fig. 7 Interface for the list of blocks in TweetDeck

The last example of the application in this paper is an application from Blogger [8]. This application allows user to create a professional-looking blog without knowing or writing the programming code. Fig. 8 shows the interface of Blogger which the red-dotted box represents the block space. Same as in the second example, the blogger also has the interface for the available listed block as shown in Fig. 9 which was named as Gadget.

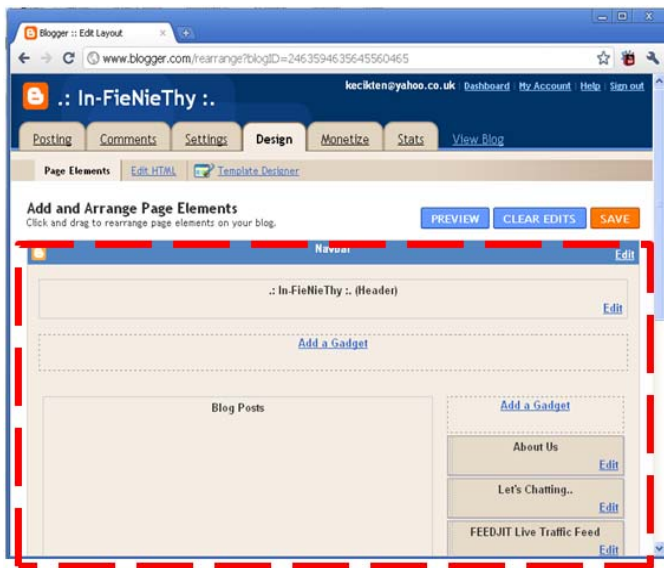


Fig. 8 Interface of the Blogger

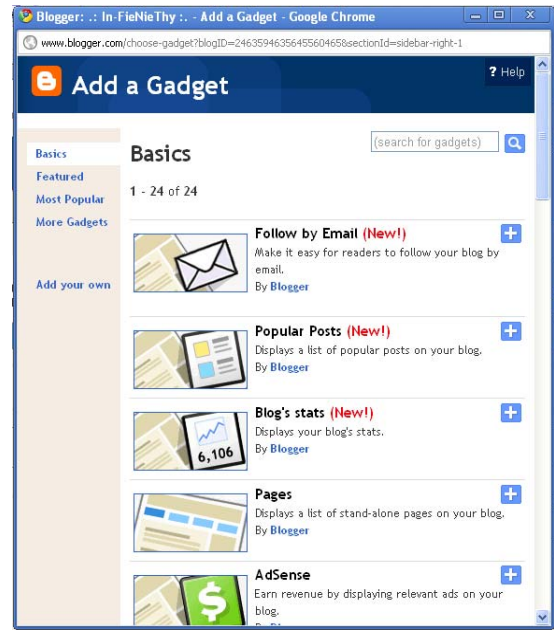


Fig. 9 Interface for the list of blocks in Blogger

The comparison of Google Earth, TweetDeck and Blogger is shown in Table I. The comparison is based on several criteria and metrics which are used to measure the effectiveness and efficiency of the programming effort in implementing the application. Such criteria and metrics are speed development, design quality, usability, reusability, block customization, user acceptance, application platform and range of user. The evaluation of the comparison of these three applications examples which were believed to apply the block-based programming approach to implement the application, are based on user's first expression while trying to create an application from the available selected blocks.

TABLE I
COMPARISON FOR THE THREE EXAMPLES

	Google Earth	TweetDeck	Blogger
Speed Development	Moderate	Fast	Moderate
Design Quality	Take time to understand	User Friendly	User Friendly
Usability	Maps	Social Network	Blogging
Reusability	Yes	Yes	Yes
Block Customization	Hard to understand	Easy	Easy
User Acceptance	Bad	Moderate	Good
Application Platform	Plug-in	Personal Browser	Any Browser
Range of User	Strict only for certain user	Strict only for certain user	Wide range of user

There are two types of developers which are *block developer* and *application developer*. Block developer acts as a programmer and responsible to create the available blocks

for the application. While, the application developer acts as the end-user who is responsible to create an application from the list of the available blocks. Unfortunately, the programmers do not know the particulars of the work that the end-users are trying to implement. Therefore, the programmers are struggling to meet the needs of these diverse end-users, and they tend to create hundreds of block services. Some of the challenges that the programmer might face while creating the blocks are:

- The created blocks might not be able to satisfy all the end-users needs.
- Programmer needs to understand the code from the vendors which could be hard to understand due to the implementation using different programming languages.
- The services might not be made available by the vendors, therefore the block for the unavailable services cannot be created and this will cause failure to meet the end-user's needs.
- Some of the code from the vendors will change from time to time and it needs to be updated manually by the programmer.
- Programmer also needs to provide a high security assurance to convince the end-user to use the application; due to the secret information such as password will be given to the program by the end-user.

The end-user will gain most of the benefits from using the application which implemented with block-based programming approach. After programmer had developed an application with various lists of available blocks, the work of the end-user would become much simpler because they could add the functions or services he or she wanted from the list of available blocks. The challenges that the end-user might be face are the risk of giving out the private information such as password to the application in order to use the services. Besides that, since the appropriate metaphors have some of their capabilities and limitations which diverge commonly depending on the users and their purposes, the end-user need to learn the meaning of some metaphors.

Table II in the following shows the benefits of the block-based programming approach from the perspectives of both the programmer and the end-user of the application program.

TABLE III
BENEFITS OF BLOCK-BASED PROGRAMMING APPROACH

Block Developer/	Application Developer/
------------------	------------------------

Programmer	End-User
<ul style="list-style-type: none"> • Instead of developing the services, programmer can borrow the code from the available vendor in order to use and provide the services to the end-user. • The approach will cause in high decreasing of the associated application development costs due to less number of programmers need to develop the application. • The available API can be included into the application without approval from anyone. • The approach is able to satisfy most of the user's need. 	<ul style="list-style-type: none"> • The application allows the selected services or functions to appear which the end-users want to use or see. • The end-user not necessarily required to have professional IT skills. • The existing application is made possible to be reused. • The end-users are freedom to observe and create the application without need to bother the programming part of the application.

IV. CONCLUSION AND DISCUSSION

In this paper, a block-based programming had been elaborated in term of it principles, main characteristics, structure and the sketch design of the approach. The block-based programming approach is a combined approach of the component-based programming approach with end-user programming paradigm. Also discussed in this paper are the benefits of block-based programming approach contribute to the community and also the challenges of using block-based programming approach. Three applications examples are evaluated and compared in term of their development speed, design quality, usability, reusability, block customization, user acceptance, application platform and range of user. The reason of showing some of these examples is to show some real applications and analysis in term of block-based programming approach. Finally, based on the evaluation and comparison from the examples, important issues had been discussed, and the challenges and benefits had been specified.

REFERENCES

- [1] H. Mohamad, A. Patel, Y. Tew, R. Latih, Q. Qassim, *Principles and Dynamics of Block-based Programming Approach*, IEEE Symposium on Computers & Informatics (ISCI 2011), pp 340-345.
- [2] A. Ismail, M. Djasmir, N. Omar, A. M. Zin., *Designing And Implementing Blocks for Developing Educational Software for Children with Learning Disabilities*, National Conference on Programming 2009 (ATUR'09), Putrajaya, Malaysia.
- [3] 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, pp. 977-982.
- [4] (2008) IBM Software Group, *IBM WebSphere sMash Version 1.1*. IBM Corporation United State of America, [Online]. Available: <http://www-01.ibm.com/software/web servers/smash/reqs/>.
- [5] Ian Sommerville, *Software Engineering Eighth Edition*, Addison-Wesley, pp. 443, 2006.
- [6] Google Earth, [Online]. Available: <http://www.google.com/earth>
- [7] TweetDeck, [Online]. Available: <https://www.tweetdeck.com/desktop>
- [8] Blogger, [Online]. Available: <http://www.blogger.com/features>