

A Block-Based Storytelling Approach for End-User Software Development

Abdullah MohdZin, Marini Abu Bakar and Hamizah Mohamad Hariri
Faculty of Information Science and Technology,
Center for Software Technology and Management,
Universiti Kebangsaan Malaysia, Bangi Selangor, Malaysia

Abstract: Storytelling is a process of conveying events by words, images and sounds. It has been used in every culture as a means of entertainment, education or for instilling moral values. The power of storytelling is now being realized by computer science and software engineering community. It has been argued that storytelling will enable the actual requirement to be identified earlier in the software development process. A number of software development environment based on story telling approach have been developed such as Storytelling Alice, Scratch and JPie. This approach has also been identified as one of the most suitable approach for end-user software development. This study describes the design and implementation of a blocks integration tool for Block-Based Software Development by using storytelling approach. Block-based software development is a new approach that support end-user software development. A block is basically a single-layer software component that can perform a specific function. It is assumed that in this software development environment, there will be a large number of blocks available. End-users can develop applications by selecting, customizing and combining blocks.

Key words: Storytelling approach, blocks integration, end user development, community, image

INTRODUCTION

Block-Based Software Development (BBSD) approach is a new software development approach to support end-user software development (Zin, 2011). A block is a single-layer software component that can be used to perform a specific task. Single-layer implies that a block cannot have sub-blocks and it cannot be a sub-block of another block. Examples of blocks for educational purposes are described by Ismail *et al.* (2009) while examples of blocks for e-Commerce development is described by Bahari *et al.* (2011).

The BBSD process model is shown in Fig. 1. In this process model, end-users develop applications by selecting, customizing and integrating software blocks. Blocks development is carried out by professional programmers.

In order to support BBSD, a number of software tool is required. One of the software tools is called blocks creation tool to support the process of blocks development (Djasmir *et al.*, 2012). Another tool needed is the blocks integration tool so support end-users in the process of selecting, customizing and integrating blocks. One of the block integration tools has been described and implemented by Sarif *et al.* (2011) that is based on visual

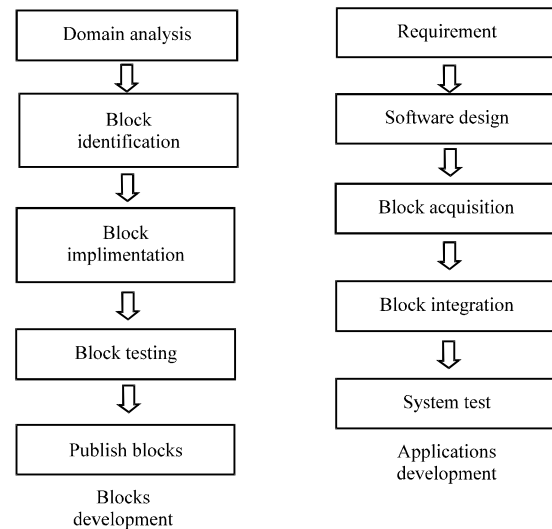


Fig. 1: Block-Based Software Development Process Model (Zin, 2011)

computing paradigm. This study describes the design and implementation of another block integration tool that is based on storytelling approach. It has been argued that a block integration tool based on storytelling is more

convenient for most of the end users. The initial idea for using storytelling in application development has been described earlier by Hariri *et al.* (2011).

Storytelling is a process of conveying event by words, images and sound. It has been used in every culture as means of entertainment, education or for installing moral values (Ozhan *et al.*, 2006; Pellowski, 1990). In education, the use of this technique makes the learning process fun, interesting and also encourages students to learn more by themselves (Norizan, 2003; Savita and Shafee, 2009). Storytelling can be classified into traditional storytelling, digital storytelling (Xin and Chin, 2010) and interactive storytelling (Cavazza *et al.*, 2003).

The power of storytelling is now been realized by the computer science and software engineering communities. It has been argued that storytelling enables the actual requirement to be identified earlier in the software development process. This approach has also been identified as one of the most suitable approach for end-user software development.

Literature review: A number of software development environment based on storytelling approach have been developed. In this study, researchers look into three examples: Storytelling Alice, Scratch and JPie.

Storytelling Alice: Storytelling Alice is a programming environment that introduces end-users (in this case middle school student) to programming as a way to create short 3D animation stories. To enable users create a story, Storytelling Alice provides high level animation that enables users to indicate interaction between characters, gallery of 3D characters and scenery and tutorial based on story example. In Storytelling Alice, users create programs by dragging and dropping code elements to become a complete narrative story. This techniques offer students an exciting way to write their narrative and at the same time both learning and writing programming codes (Kelleher, 2006).

Scratch: Similar to Storytelling Alice, Scratch allows users to manipulate media to create their own stories as a tool in learning programming language. Furthermore, Scratch also enables users to create games, interactive stories, animation, music and art and share it with other users by uploading to Scratch server as end product. To create program, users can simply drag and drop command blocks of code and then stack these blocks together to form coding scripts (Resnick *et al.*, 2009).

Table 1: Comparison between Storytelling Alice, JPie and Scratch

Comparison	Storytelling		
	Alice	JPie	Scratch
Drag and drop approach for writing program	Yes	Yes	Yes
Provides the option to write the program	No	Yes	No
Drop-down menu approach for writing program	Yes	No	Yes
Using natural languages	Yes	No	Yes
Provides 3D animation or image object	Yes	No	Yes
Provide tutorial	Yes	No	Yes
Provide mechanism or tool for compiling	Yes	Yes	Yes
Ability to store and run file in different formats	No	No	Yes

JPie: JPie is a programming environment that enables end-users to develop program using Java programming language. JPie treats programming as an application in its own right, providing a visual representation of class definitions and supports direct manipulation of graphical representations of programming abstractions and constructs. JPie’s visual unit is the capsule. Capsules represent variable declarations, variable accesses, properties, methods, method calls, constructors and constructor calls and can also contain constants and expressions. Every capsule has a textual identifier, an icon to indicate type and a colour to indicate scope. Users create program by manipulating capsules and other objects within identified semantic regions (Birnbaum and Goldman, 2005).

Comparison: Table 1 compares the facilities and features provided by three applications described earlier.

MATERIALS AND METHODS

The method used for developing the IDE consists of the following steps:

- Creating a model for block-based storytelling
- Creating the specification language for block-based storytelling
- User interface design
- Use case diagrams
- Sequence diagrams

Each of these items is explained in the study.

Model for block-based storytelling: A number of models for storytelling approach have been proposed. The proposed model is based on the linear storytelling approach proposed by Spierling (2005). The model, shown in Fig. 2 is normally used in the process of creating

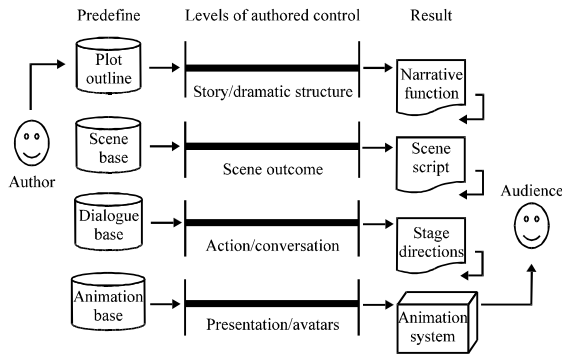


Fig. 2: Linear Storytelling Model (Spierling, 2005)

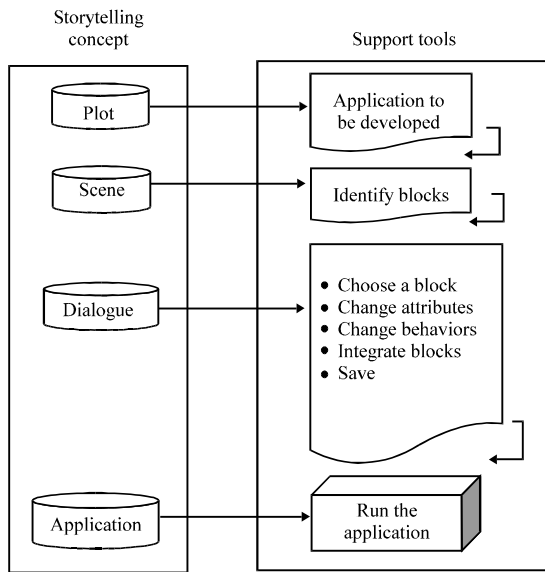


Fig. 3: Model for Block-Based Storytelling approach

animated films. The model for the block-based storytelling is shown in Fig. 3. Explanations for each item in the model are as follows:

- Application to be developed: an over view description of the application to be developed by users
- Identify the required blocks: based on the idea of the application to be produced, user identified the blocks needed
- Choose block: user chooses a block to be customized from the list of blocks that have been identified
- Customized the desired attributes: users customize attributes of the block to suit their needs
- Modify block behavior: user modifies the behavior of the block
- Block integration: user integrates the first block with the following blocks using integration technique to produce a continuous application

Table 2: Specification language of Block-Based Storytelling tool

Task	Proposed language
Identify the required block	
Select block	Select <block name>;
Modify attributes	Change <attribute> to <new attribute value>; Edit <attribute> to <new attribute value>;
Modify behaviour	Change <behaviour> to <new behaviour value>; Edit <behaviour> to <new behaviour value>;
Block Integration	Connect <block name> to <next block><integration type>;
Save	
Run	

- Save the story file: the blocks that are integrated must be run to see the product formed. If the product formed was perfect as needed, the program will be saved as a complete application

Specification languages for block-based storytelling: The description of the proposed specification language used in the Block-Based Storytelling tool is shown in Table 2.

User interface design: Figure 4 shows the user interface design of Block-Based Storytelling tool. The user interface includes the following:

- Menu bar: consists of file menu that contains new file, open file, save, save as and exit, edit menu, import menu, run menu and help menu
- Block space: display all the blocks imported by the user and file application that had been created, in tree structure
- Method space: display some of the methods for user reference in writing the story for the application
- The workspace: space to write the story. In this space, user will create, edit and integrate each of the block using storytelling approach to form the complete application

Use case diagram: Figure 5 shows the use-case diagram for Block-Based Storytelling tool. The six functional requirements are as follows: Create Project, Import Blocks, Select Blocks, Customize Blocks, Integrate Blocks, Run Project and Save as Project.

The detail scenario for create project, integrate block and run project are described in Table 3-5, respectively. There are three types of Block Integration Method namely randomly order, sequential order and condition order integration method:

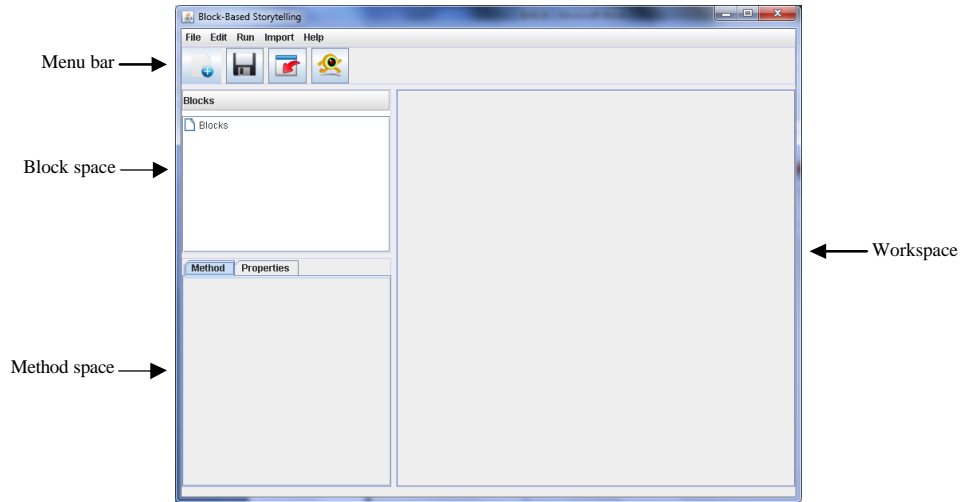


Fig. 4: User interface design

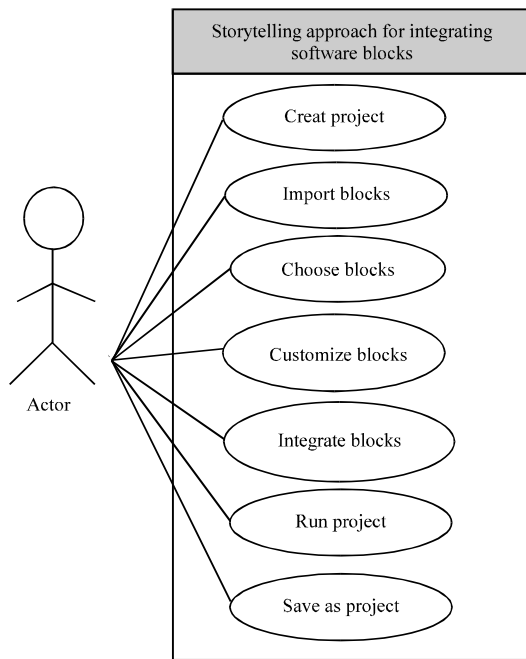


Fig. 5: Use case diagram for Block-Based Storytelling approach

- Randomly order (non-deterministic): after first block are executed, system will randomly choose the next block to be execute
- Sequential order: after first block are executed, system will execute the next block determined by the user
- Condition order (deterministic): after first block are executed, depending on the condition chosen by user, system will select the next block to be executed

Table 3: Use case scenarios for create project

Name of use case Create project	
Explanation	User needs to build the project to determine the name and location for storing work files to be developed
Actor	End user
Priority	Apply
Pre-conditions	Project isnot yet created
Post-conditions	File appears on workspace for user to write a program. Java file created
Trigger	User wants to develop an application
Sequence of events	Users click the 'New Project' button or <File><New Project> menu at the menu bar Tool issued a 'New Project' window User fill project name User click 'Browse' button to select directory to save the file Directory name set automatically at project location User click 'Ok' button Tools close the 'New Project' window Tool creates Java file based on the named given Tool show a file at workplace using named given by user

Table 4: Use case scenarios for integrate block

Name of use case Integrate block	
Explanation	To integrated the block created with each other so that it will be a continuous application To integrate the block, user needs to specify the type of integration
Actor	End user
Priority	Apply
Pre-conditions	Block has been imported, selected and customized
Post-conditions	A sentence to integrate block are produced
Trigger	User wants to integrate the selected blocks
Sequence of events	User write integration sentence by indicating the type of integration such as 'join block A-D randomly' means of connecting block A-D at random

Sequence diagram: The sequence diagrams are used to show how process operates with one another. Figure 6-8 show the sequence diagram for create project,

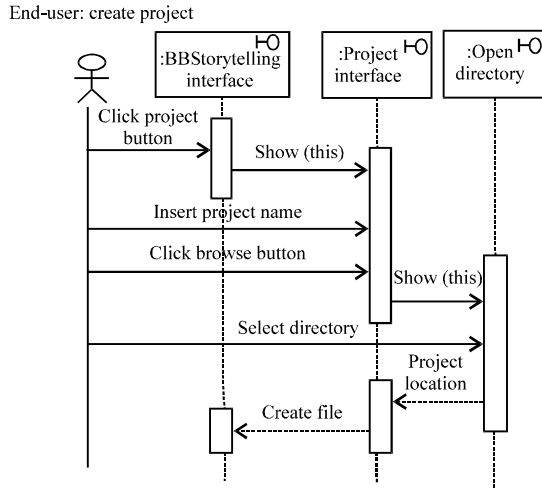


Fig. 6: Sequence diagram for create project

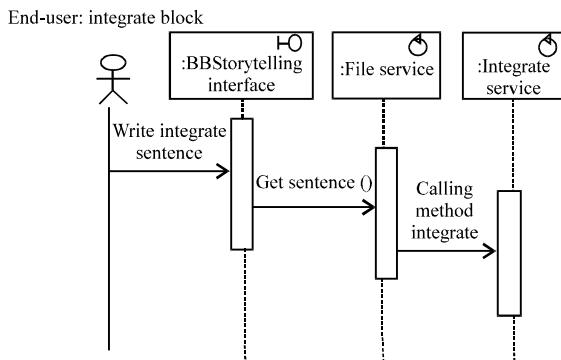


Fig. 7: Sequence diagram for integrate block

Table 5: Use case scenarios for run block

Name of use case	Run block
Explanation	To run the blocks that have been modified and integrated as an application If the application is not as required, the user can edit the integration sentence
Actor	End user
Priority	Apply
Pre-conditions	Block has been selected, imported, modified and integrated
Post-conditions	The complete application in the form of .exe file
Trigger	Users wish to implement and run the application
Sequence of events	User click 'Run' button Tool save the program Tool compile and run the programming code Tool run the created application User determine whether the application fulfil the requirement User re-customized the program in the workspace

integrate blocks and run blocks, respectively. The sequences of events are described in the use case scenario in previous study.

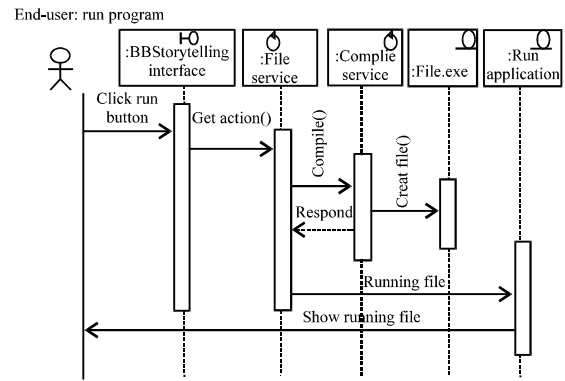


Fig. 8: Sequence diagram for run program

RESULTS AND DISCUSSION

The block integration tool based on storytelling approach is implemented using Java programming language. Steps to use the software tool can be described as follows. In this example, researchers use the blocks from the educational domain that are described by Ismail *et al.* (2009).

First step: Create a new file by clicking on the <File><New Menu> menu. This action will create a new file named as given by the user at the workspace area.

Second step: User can import blocks using the <Import> menu. Once the blocks are imported, all available blocks will be shown. User can add any other block that is compatible with the application that they wanted to create. Figure 9 shows the layout of the blocks.

Third step: Write a sentence (in story form) to select and edit the block in the workspace. The user needs to write sentences based on the methods available in the method space. To speed up the process, the user can just click on the suitable method in method space. The selected method will appear in the workspace. The user then needs to fill up the required words to complete the sentence.

Fourth step: After creating and editing the block, the user needs to integrate each of the blocks with another block. The user can choose to integrate the blocks using sequential order, random order or condition order. Figure 10 shows an example of a complete 'Learn ABC' application using the storytelling tool.

Fifth step: Run the file. User can still make changes required in the workspace.

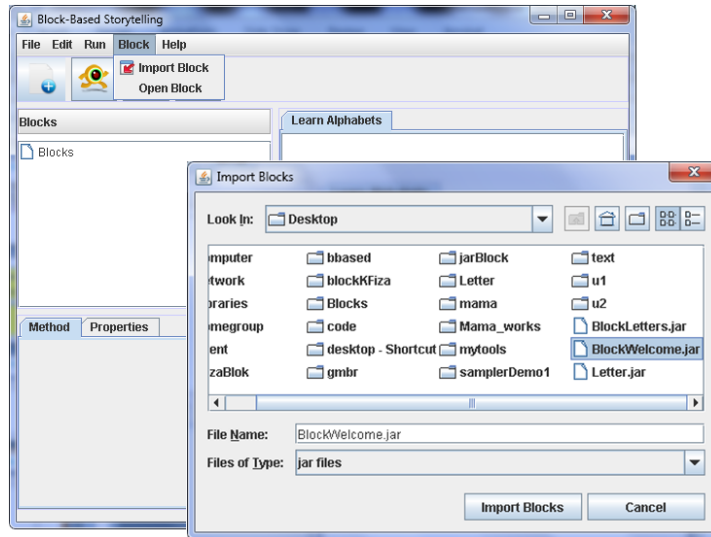


Fig. 9: Import blocks

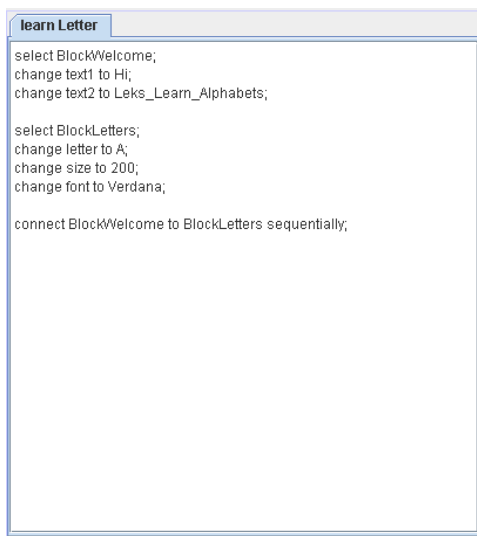


Fig. 10: Example of 'learn ABC' through storytelling

Last step: Save the file as 'LearnABC. bbp' which is in zip file format. This is the complete application file.

Evaluation: Evaluation of the block-based storytelling was carried out in order to obtain users' feedback on the appropriateness of the proposed tool and the use of storytelling approach. Six respondents were selected in this evaluation process. They were briefed on the concept and the construction of the tool. They were then demonstrated on how to use the tool. Respondents were required to use the tool to develop a learning application using the blocks provided. Upon completion the

application development, respondents were asked to fill in the evaluation form to determine the level of satisfaction and perception of the user on the tool. The evaluation form consists of three parts: personal information, user interface design and the ease on interaction while using the tool to integrate blocks.

As a result of the evaluation, it is found that all of the respondents agreed that the effectiveness of storytelling approach is good and the process of application development using this approach is easy to be understood.

For the user interface design, all of the respondents felt that the interface of this tool is easy to be used. Four of respondents agreed that the symbols used are easily understood and appropriate while the remaining two said there were symbols of buttons and menu bars that are less appropriate.

All respondents agreed that the tutorials and instructions for using the tool help users to develop applications. They also agreed that they could learn and understand the use of the tool without much problem with half of them did not need to refer to instructions or tutorial to develop the applications. Five of the respondents thought that all the functions provided are appropriate to assist users to develop applications and users do not know whether each of the remaining functions are appropriate or not.

CONCLUSION

In this study, researchers have described the design and implementation of a software tool for integrating

blocks by using storytelling approach. The first part of the study describes the concept of storytelling in software development followed by a description of a model for applying Storytelling approach in block-based software development. The rest of the study describes the design, implementation and evaluation of the tool. The availability of this tool enables end users such as parents or teacher to edit and integrate blocks to develop a software application. Based on the evaluation, it seems that end users consider storytelling approach is easier to use. Apart from that, this approach can also train users to think logically and hence can help them improve their ability to develop software applications.

REFERENCES

- Bahari, R., A.M. Zin and S. Idris, 2011. Blocks identification and implementation for e-Commerce development tool. *Asian J. Inform. Technol.*, 10: 327-334.
- Birnbaum, B.E. and K.J. Goldman, 2005. Achieving flexibility in direct-manipulation programming environments by relaxing the edit-time grammar. *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, September 20-24, 2005, IEEE Computer Society Washington, DC., USA., pp: 259-266.
- Cavazza, M., F. Charles and S.J. Mead, 2003. Interactive storytelling: From AI experiment to new media. *Proceedings of the 2nd International Conference on Entertainment Computing*, May 8-10, 2003, Carnegie Mellon University Pittsburgh, PA, USA., pp: 1-8.
- Djasmir, M., S. Idris, M.A. Bakar and A.M. Zin, 2012. An integrated development environment for blocks creation. *Asian J. Inform. Technol.*, 11: 194-200.
- Hariri, H.M., M.A. Bakar and A.M. Zin, 2011. Story telling approach for integrating software blocks. *Proceeding of the International Conference on Electrical Engineering and Informatics*, July 17-19, 2011, Bandung, Indonesia, pp: 1-5.
- Ismail, A., N. Omar and A.M. Zin, 2009. Developing learning software for children with learning disabilities through block-based development approach. *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics*, August 5-7, 2009, Kuala Lumpur, Malaysia, pp: 299-303.
- Kelleher, C., 2006. *Motivating programming: Using storytelling to make computer programming attractive to middle school girls*. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh.
- Norizan, M.D., 2003. *Pendekatan bercerita dan permainan dalam pembangunan perisian kursus Akhlak Islamia*. Master Thesis, Universiti Kebangsaan Malaysia (In Malay).
- Ozhan, T., A. Gunefler, E. Ongun, A. Demirag and O. Koroglu, 2006. Using storytelling in education. *Proceedings of the International Symposium of Interactive Media Design*, April 28-30, 2006, Yeditepe University, Istanbul, Turkey, pp: 1-4.
- Pellowski, A., 1990. *World of Storytelling*. 3rd Edn., H.W. Wilson, Bronx, New York.
- Resnick, M., J. Maloney, A. Monroy-Hernandez, N. Rusk and E. Eastmond *et al.*, 2009. *Scratch: Programming for everyone*. *Magaz. Commun. ACM.*, 52: 60-67.
- Sarif, S.N., S. Idris and A.M. Zin, 2011. The design of blocks integration tool to support end-user programming. *Proceedings of the International Conference on Electrical Engineering and Informatics*, July 17-19, 2011, Bandung, Indonesia.
- Savita, K.S. and K. Shafee, 2009. An exploratory study of storytelling approach as an instructional tool from educators perspective. *Proceedings of the International Conference on Computer Technology and Development*, Volume 2, November 13-15, 2009, Kota Kinabalu, pp: 480-483.
- Spierling, U., 2005. Interactive digital storytelling: Towards hybrid conceptual approach. *Proceeding of Digital Games Research Conference 2005, Changing Views: Worlds in Play*, June 16-20, 2005, Vancouver, British Columbia, Canada, pp: 1-11.
- Xin, L. and H. Chen, 2010. The application of digital storytelling in teaching evaluation. *Proceedings of the 2nd International Conference on Education Technology and Computer*, Volume 4, June 22-24, 2010, Shanghai, pp: 249-252.
- Zin, A.M., 2011. Block-based approach for end-user software development. *Asian J. Inform. Technol.*, 10: 249-258.