An Online Metadata-Driven Editor for Rich Maze Video Games for Education

DESSISLAVA VASSILEVA, NIKOLAY PENCHEV
Department of Software Technologies,
Sofia University "St Kl. Ohridski"
5, J. Baurchier Blvd., 1164 Sofia,
BULGARIA
ddessy@gmail.com, nikolay.pen4ev@gmail.com

Abstract: - In recent years, video games have become increasingly popular and used not only for entertainment but also in the form of serious games in the process of education and vocational training, in advertising campaigns, etc. However, serious games still cannot be used widely by teachers and instructors mainly because of their high-cost development and difficulties in maintaining educational content and integration of different pedagogical strategies in the game depending on training goals. Therefore, there is a need for software platforms for design and generation of video games for educational purposes from trainers and pedagogues. In this article, we present an open software platform that allows non-IT professionals like teachers, pedagogues, and educationalists to construct and generate rich educational video maze games. The platform is under development in the scope of the APOGEE (smArt adaPtive videO GamEs for Education) research project and includes an online, metadata-driven editor for design of rich educational maze games. The designed games can be exported in the form of an XML document including the semantic structure of game, didactic content and audio-visual interiors of maze halls. The editor is controlled by a predefined XSD schema which is used for user-input validation. The editor's purpose is to remove the complexity for non-technical people and to allow them easily to build an XML document of maze configuration that will be imported into the Unity 3D environment. Hereby, non-IT professionals will be able easily to create and generate new rich educational maze games incorporating targeted educational content and pedagogical strategies.

Key-Words: - educational games, editor, XSD-driven, survey, APOGEE

1 Introduction

For the past two decades, video games have become a popular and effective tool not only for entertainment but also for supporting a range of activities in areas such as education, vocational training, rehabilitation, advertising, production and much more. Computer games of this type are called serious (or applied) [1] and continue to be used more and more in many spheres of human life [2, 3]. However, the serious games are still not an affordable means for game-based learning (GBL) in schools and universities [4] due to some strong problems with the construction of such games, especially their high development costs, less attraction than entertainment games, difficult match of learning mechanics to the gaming mechanics, and need of inclusion of educational and pedagogic paradigms in the gameplay [5].

The problems with construction of serious games for education lead naturally to creation of software platforms for design and generation of such video games [4, 6]. The paper presents an online, metadata-driven editor for creation of rich educational maze games. The editor for creation of rich educational maze games is developed in the scope of the APOGEE (smArt adaPtive videO GamEs for Education) research project[1]. The open platform of APOGEE will allow non-IT professionals like teachers, pedagogues, and educationalists to construct and generate rich educational video maze games. A rich educational maze is defined as a maze 3D video game with multimedia learning content tailored upon the player/learner model attributes such as demographic characteristics, demonstrated outcomes, learning style and emotional status [7]. Maze tailoring involves not only the game features but the didactic content, as well.

[1] http://apogee.online/index-en.html

The maze generation process is based on a formal game description designed for the moment manually as an XML document applying semantic structuring of both game and didactic content. The online editor will allow teachers and pedagogues to create their own educational games just by defining maze connectivity, audio-visual interior of the maze halls including various puzzle mini-games, and the learning content for these games. After creation of a maze game, the editor will allow the user to validate it and to generate the XML description in order to be used by the APOGEE Maze Builder platform [4] for an automated generation of a rich video-maze for the Unity 3D environment. Thus, the editor will facilitate non-ICT professionals in automatized construction of serious video games for education.

2 Problem Formulation

In last decade, serious games have been identified to suffer from serious problems such as high production costs, less attraction than games for fun, lack of framework for combination of learning mechanics and gaming mechanics, and different gameplay options reflecting educational and pedagogic paradigms [5]. Along with the decline of the serious game industry, several alternative approaches appeared concerning automatic or semi-automatic construction of educational games. Several customizable platforms have been proposed for the automatic creation of educational games [8].

The present approach is based on the maze generator proposed within the scope of the ADAPIMES research project[2]. Based on Python script templates, the generator was able to create 3D adventure mazes with puzzles for unlocking doors, with customizable maze structure and personalized content according to the player model characteristics. Next to ADAPTIMES, a Maze Builder was suggested [4] where game creators developed an XML description of the maze structure and hall environment by means of a complex template including learning content, puzzle mini-games embedded into the maze halls, and audio-visual assets. Along with the practical experiments, there was identified the need of a graphic maze editor able to generate and validate the complex XML structure based on maze design done by teachers and other non-ICT domain specialists.

Another problem of a maze design guided by the editor was identified to be imposed by future eventual changes in the XML structural description of the maze, e.g. by adding more learning boards and additional puzzle types. In order to reflect such changes, the editor must be rewritten and deployed again on the game portal. However, such a solution takes a lot of efforts and time, therefore is not appropriate. On the other side, if the editor will be driven (controlled) by an XML schema [9] (responsible for the correct game description in form of XML file), the future enhancements and updates in that schema will be automatically reflected by the editor as far as the tool reads each XML element or attribute definition and represents it online. Examples for similar XSD-driven editors are the RAGE advanced editor for game assets metadata [10], EDI – a template-driven metadata editor for research data [11], and the illiterate editor applied for a metadata-driven revert detection in Wikipedia [12].

3 The APOGEE Maze Editor

The APOGEE maze game design editor is an online tool dedicated to an easy and intuitive design of rich educational mazes. Maze design makes an essential part of the overall APOGEE game construction process.

3.1 The APOGEE Maze Game Construction Process

The game construction process is presented in fig. 1. Game designers can define the game either formally by writing an XML document presenting both the learning and gaming contents using a template or can use the online maze editor for defining the maze game by a simple method. Bontchev and Panayotova (2017) found that only a third of the teachers could construct XML documents for their games, hence, the project team develops an online maze editor for facilitating the maze design. The editor is controlled by the maze XML Schema (i.e., an XSD document) in order to reflect future changes in the definition of the maze game. It generates an XML document describing the maze game, that is submitted to a maze builder together with all the multimedia content needed for the game. The maze builder generates the maze game for Unity, thus, the maze can be built for different target platforms. In fig. 1, all the modules presented in dotted line are under development.

3.2 The APOGEE Maze Game Design Editor

The APOGEE maze game design editor is going to be integrated within the overall APOGEE gaming platform as represented in fig. 1. Authorized users

[2] http://adaptimes.eu/

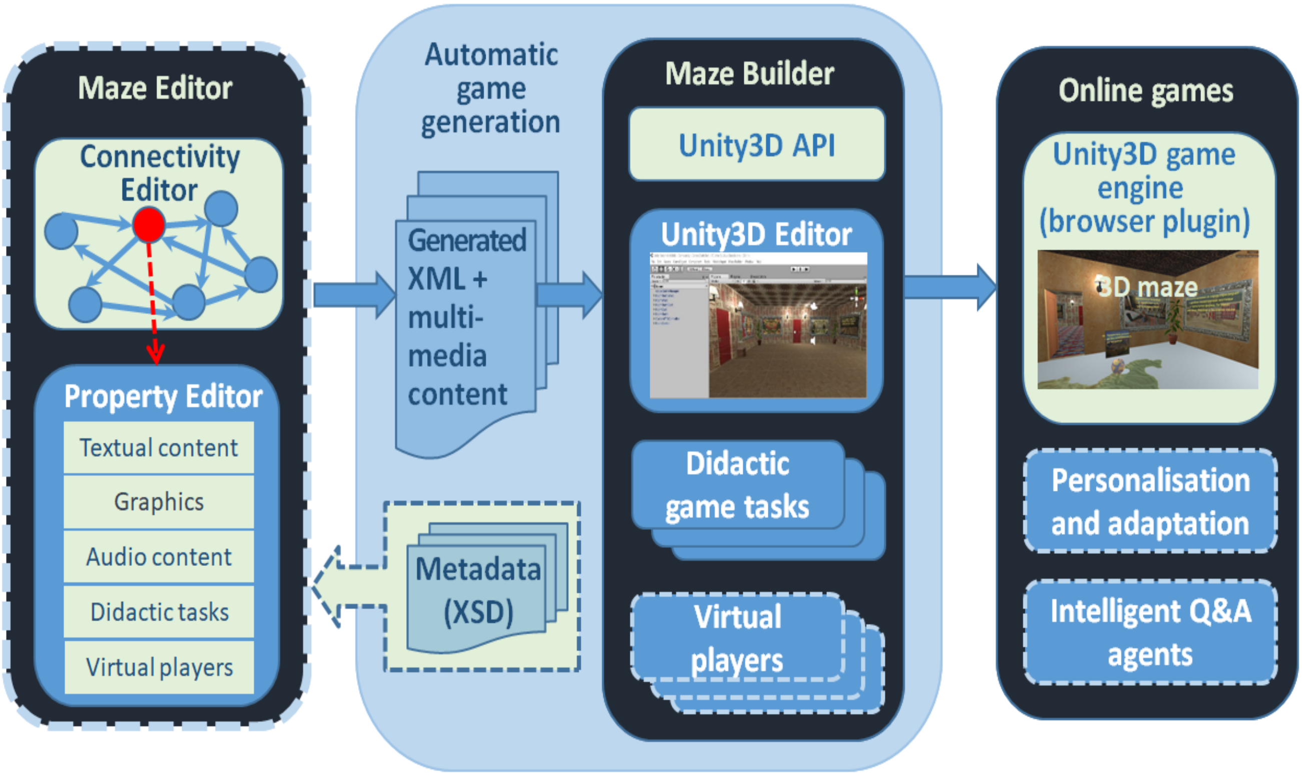


Fig. 1 The construction process of APOGEE rich maze games

such as teachers, pedagogues and educationalists are offered to create a new rich maze game, to modify their existing games, and to upload their multimedia game assets (recourses) such as textures and 3D objects to the platform database. The assets can be saved to the database as private resources (belonging only to the person uploading them, i.e. to their owner) or as public ones (available to be used by all the authorized users). After uploading all the needed assets, the user can enter the editor and continue with the design of a new or already existing game.

The maze editor consists of a maze connectivity editor, maze interior editor, maze validator, and XML generator.

3.2.1 The Maze Connectivity Editor

The maze connectivity editor allows the user to create and update the connectivity graph of the maze. Mazes have a planar graph, where a graph node represents a hall/room of the future maze and an arc signifies a door between two halls. Each node may have up to four doors to its neighbour nodes. For simplifying the maze game generation process, the connectivity graph of the maze is restricted to a connectivity grid, where two nodes (maze halls) connected by a door should always have a common wall between them.

Fig. 2 represents a sample maze connectivity grid in the editor. The maze creation process starts with the initial hall show in the figure as a dark cell having coordinates (0, 0). Each newly created node without neighbours is shown with four bright arrows – up (north), right (east), down (south), and left (west). When the user clicks onto one of the next cells in the grid, a neighbour node is created and all the bright arrows leading to it from existing halls are replaced by dark ones. At the maze presented in fig. 2, three doors connect the starting hall (0, 0) to the halls (0, 1), (0, -1), and (-1, 0), while another door leads from its right neighbours (1, 0) to (0, 0). For the node (3, 2), there is only one inbound door from (2, 2) and no outbound doors. At this point, the user can continue creating new halls reachable from the hall (3, 2) by clicking on the cells (3, 3), (4, 2), or (3, 1). However, if the cell (4, 2) will be clicked, a new node (4, 2) will be created without to be connected with any neighbour node (because not having a shared wall with any of the neighbour cells).

The maze designer can delete maze nodes, whereupon they disappear from the connectivity grid and the doors leading to them are presented by bright arrows. The starting node/hall cannot be deleted. As well, the designer can change a door in one of the following ways:

- A door can be reversed, i.e. its direction can be changed to be the opposite one;
- A door can be set as bidirectional (in the generated XML such a door is presented as one inbound and one outbound door, and in

the generated maze it will be shown as an empty frame in the wall);

- A door can be deleted, whereupon the generated hall will be shown as having a wall without any doors.

If the user deletes a node which breaks the connectivity of the maze, the editing and saving are frozen until the user provides a valid connection between the nodes. We have a revert button for the connectivity editor, which keeps only the last saved stage, but once deleted and saved, all the deleted data will be lost. The maze connectivity editor gives the user really base view of the whole maze, which removes a lot of from the XML complexity.

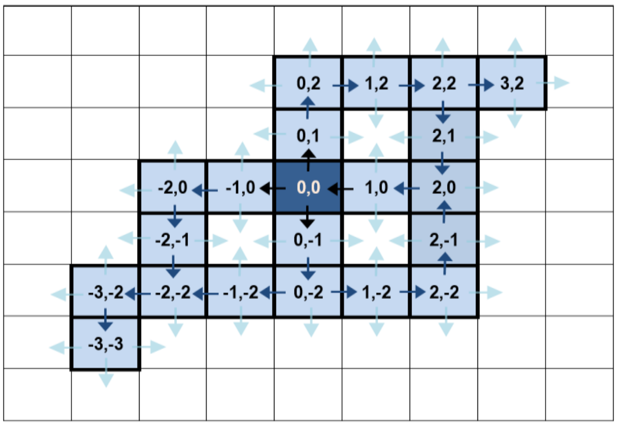


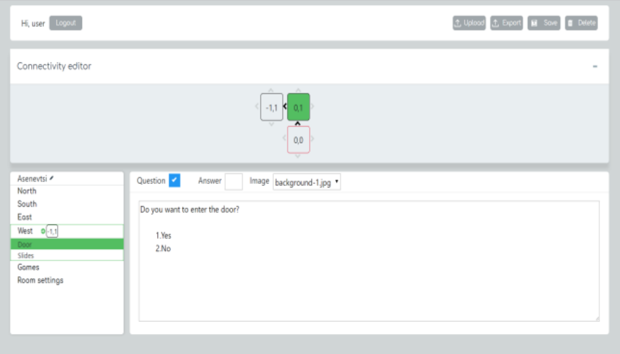
Fig. 2 A view of a sample maze connectivity graph

3.2.2 The Maze Interior Editor

The maze interior editor is a Web application where the game designer sets the description and content of each hall of the future maze. Within the maze connectivity editor, the designer selects a node/hall and then can switch to the interior editor representing this hall according to the XSD metadata description. Here, the maze designer has a user interface for setting/updating features of the maze hall, such as:

- Learning boards (called slides) having didactic content (text and/or images);
- Puzzle games of various types such as answering a question for unlocking a door, word games (word soups or quizzes), 2D image and memory puzzles, shooters, roll-a-ball puzzles, discovering hidden objects, and others. Each puzzle represents a learning task to be solved by the player, which might be mandatory or optional;
- 3D objects situated at given places in the hall;
- decorative visual elements like illumination and textures for the walls, the floor, and the ceiling;
- audio and sound effects.

All the elements info will be displayed in containers where only the selected element will be visible, so if you add roll-a-ball puzzles to the current room and select it, you will see only the elements/fields for it. For example, fig. 3 presents the question for unlocking the door of the west wall of hall number (0,1).



Question ☑ Answer □ Image background-1.jpg

Do you want to enter the door?

1.Yes
2.No

Fig. 3 Editing the question for unlocking the door of the west wall of hall (0,1)

The editor will allow storing and subsequent editing of the labyrinth by its creator, including the connectivity column and the characteristics of each node, as well as generating and validating a labyrinth XML description to be used by the Maze Builder platform for automated video game generation within Unity 3D. Hence, the benefit of such an editor would be very great, as it would be a free and intuitive tool for creating and generating descriptions of labyrinth 3D video games for learning purposes.

3.2.3 The Maze Validator

The editor's user interface is controlled by a pre-defined XSD schema applied for generation of the maze. The XSD schema is applied for validation of the XML document describing the designed maze. Thus, the XSD-based editor will allow the creation of mazes of different types by replacing the XSD scheme, that is, the metadata describing the maze.

The XSD scheme is used for limitation of the possible values of the maze parameters. During the maze building workflow, the user can add elements which are defined and applied by the scheme type validation of the user input. That gives us really smooth user-experience. The XSD schema is parsed by the editor to JSON format only for flexibility. On the other hand, the XSD scheme is applied for validation of the XML documents created locally (outside the editor) by experienced users uploaded and uploaded to the portal for a further game generation.

3.2.4 The XML Generator

The editor can generate an XML document describing the designed maze and compatible with the XSD schema applied during the maze design process. Through that XML document, the Maze Builder automatically generates 3D video-mazes in the Unity 3D environment.

All the XML documents exported by the editor are valid for uploading in Unity 3D. Every Maze is stored separately as a JSON document into the database. The XML generation is made only on export. We are using JSON format for more flexibility and maintainability.

The generated XML is a valid instance of the XML Schema provided to the editor. The following XML snippet describes one of the roll-balls-to-positions game of a sample history maze game generated by the APOGEE platform [13]:

```
<Game>
  <MinPoints>0</MinPoints>
  <GameElements>
    <GameElement>
     <Text>Място на въстанието</Text>
     <Image>flag1.jpg</Image>
     <Name>Ball1</Name>
     <Type>Ball</Type>
     <Texture>marble-green.jpg</Texture>
   </GameElement>
   <GameElement>
     <Text>Търново</Text>
     <Image></Image>
     <Type>Circle</Type>
     <Texture></Texture>
     <Ball>Ball1</Ball>
   </GameElement>
   <GameElement>
     <Text>Присъединен голям град</Text>
     <Image>flag2.jpg</Image>
     <Name>Ball2</Name>
     <Type>Ball</Type>
     <Texture>marble-green.jpg</Texture>
   </GameElement>
   <GameElement>
     <Text>Белград</Text>
     <Image></Image>
     <Type>Circle</Type>
     <Texture></Texture>
     <Ball>Ball2</Ball>
   </GameElement>
   <GameElement>
     <Text>Друго</Text>
     <Image></Image>
     <Type>Circle</Type>
     <Texture></Texture>
     <Ball>Ball3</Ball>
   </GameElement>
   <GameElement>
     <Text>Друго</Text>
     <Image></Image>
     <Type>Circle</Type>
     <Texture></Texture>
     <Ball>Ball4</Ball>
   </GameElement>
 </GameElements>
</Game>
```

Next, fig. 4 presents a view of the generated roll-balls-to-positions game inside one of the maze halls.

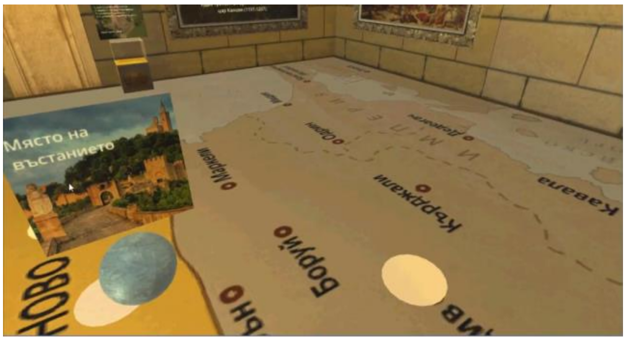


Fig. 4 A view of the generated roll-balls-to-positions game

4 Discussion

The general goal of this study was to present an online, metadata-driven drag-and-drop editor for creation of maze games that allows non-IT professionals like teachers, pedagogues, and educationalists to construct and generate rich educational video maze games. This editor provides possibilities instructors easily to create different variants of a course as each one of them is appropriate for different student groups having different level of knowledge, different learning style and/or training goals.

The APOGEE online maze game design editor does one step ahead in introducing more intensive use of serious games in the learning process because many researchers have found that one of the reasons for slowing down this process is the need for educational game creators to have high-level IT knowledge and skills [4]. Another disadvantage in the current state of educational games is high-cost development of a game that embedded a specific pedagogical strategy [5]. The presented game editor allows easily to be constructed a serious game incorporating a specific educational strategy. Then, this game can be used in a new one by retaining part of educational and gaming content and modifying the rest (for instance changing the content of some learning units, learning activities, game assets, halls interior, etc.) in order to be implemented other pedagogical strategy. Each one of constructed variants of a game can be used for different student groups (that have different learning goals, learning styles or different level of knowledge). Thereby, it can accomplish a static adaptation of the training content that is one of the crucial factors for an effective learning process.

The challenges that resolve the APOGEE editor make it very promising online maze game design editor and our next research will be focus on conducting an experiment with teachers, instructors, and pedagogues for evaluation of the usability of the platform.

5 Conclusion

The paper presented the APOGEE online maze game design editor dedicated to easy and intuitive construction of 3D adventure educational mazes. It consists of four main components - a maze connectivity editor, a maze interior editor, a maze validator, and XML generator. The maze connectivity editor allows non-IT people to design mazes with different forms including halls connected with each other through unlocking doors. In the maze, interior editor instructors using drag-and-drop tool can define learning units with personalized content according to the player model characteristics and the chosen pedagogical strategy, puzzle mini-games embedded into the maze halls, and audio-visual assets to represent the learning content. The user interface of the maze game design editor is based on a pre-defined XSD schema and all components of a maze game (structure of the maze, maze halls interior, connection between themselves, mini-games and audio-visual assets in each hall, etc.), connections and relationships between them are described as metadata in an XML document that can be exported. In the game generation process, the pre-defined XSD schema is used for validation of the exported XML document presenting the maze. Further, for game generation can be used an external XML file that is not exported by the APOGEE maze game design editor if it is compatible with the editor XSD schema.

Thus, the APOGEE rich educational maze editor addressed some of the greatest challenges of serious games [4, 14]. It is supposed to provide rich gaming and learning experience. The benefit of such an editor would be very great, as it would be a free and intuitive tool for creating and generating descriptions of labyrinth 3D video games for learning purposes.

In order to increase the efficacy of game-based learning process in the future version of the APOGEE platform can be implemented intelligent virtual players and applied a dynamic, player-centric adaptation of both difficulty of learning tasks and the audio-visual properties of the game environment. As future works, we plan to conduct practical experiments with the developed editor aimed at the construction of experimental video mazes for education, with generating XML descriptions for the particular video game. The experiments will include evaluation of the usability of the platform by pedagogues and other non-IT specialists, together with an analysis of the results obtained from a field trial.

Acknowledgements

The research leading to these results has received funding from the APOGEE project, funded by the Bulgarian National Science Fund, Grant Agreement No. DN12/7/2017.

References:

[1] Marsh, T. Serious games continuum: Between games for purpose and experiential environments for purpose. *Entertainment Computing*, 2(2), 2011, pp. 61-68.

[2] Jain, L. C. *Serious games and edutainment applications* (Vol. 504). M. Ma, & A. Oikonomou (Eds.). London: Springer, 2011.

[3] Bontchev, B. Serious Games for and as Cultural Heritage, *Digital Presentation and Preservation of Cultural and Scientific Heritage*, UNESCO, Issue No. 5, 2016, pp. 43-58.

[4] Bontchev, B., Panayotova, R. 2017. Towards Automatic Generation of Serious Maze Games for Education, *Serdica J. of Computing*, Vol. 11, No 3, pp. 249–278.

[5] GALA. *GALA Roadmap*. Deliverable D1.6, Ver. 2, November 2011.

[6] Bontchev, B. Modern Trends in Automatic Generation of Content for Video Games, *Serdica Journal of Computing*, August, 2017, Vol. 10, No. 2, ISSN 1314-7897, pp.133-166.

[7] Bontchev, B. Rich educational video mazes as a visual environment for game-based learning, *Proc. of Int. Conf. on Innovations in Science and Education*, March 20-22, 2019, Prague, Czech Republic (in print).

[8] Bontchev, B. Customizable 3D video games as educational software, *Proc. of 7th Int. Conf. on Education and New Learning Technologies EDULEARN´15*, Barcelona, Spain, 6-8 July, 2015, pp. 6943-6950.

[9] Fallside, D. C., Walmsley, P. XML schema part 0: primer second edition. W3C recommendation, 16, 2004.

[10] Georgiev, A., Grigorov, A., Bontchev, B., Boytchev, P., Stefanov, K., Westera, W., Prada, R., Hollin, P., Moreno Ger, P. The RAGE advanced game technologies repository for supporting applied game development. *Int. Conf. on Games and Learning Alliance*, Springer, Cham, 2016,

pp. 235-245. DOI: 10.1007/978-3-319-50182-6_21.

[11] Pavesi, F., Basoni, A., Fugazza, C., Menegon, S., Oggioni, A., Pepe, M., Tagliolato, P., and Carrara, P. EDI–A Template-Driven Metadata Editor for Research Data. *Journal of Open Research Software 4*, no. 1, 2016.

[12] Segall, J., Greenstadt, R. The illiterate editor: metadata-driven revert detection in wikipedia. *Proc. of the 9th Int. Symposium on Open Collaboratio*n, ACM, 2013, August, p. 11.

[13] Terzieva, V., Video Games for School Education, *Proc. of the National Conference on "Education and Research in the Information Society"*, Plovdiv, June, 2018, pp. 84-93.

[14] Bellotti, F., Berta, R., & De Gloria, A. Designing effective serious games: opportunities and challenges for research. *International Journal of Emerging Technologies in Learning (iJET)*, 5, 2010.