Mistake Chasing Game - Approach for Mobile Programming Education in Nong Lam University - Ho Chi Minh City – Vietnam

Le Cong Nga¹, Pham Cong Thien¹, Quan Thanh Tho² and Jarkko Suhonen³
¹Nong Lam University - Ho Chi Minh City - Vietnam
congngale@gmail.com, pcthien@hcmuaf.edu.vn
²Ho Chi Minh City University of Technology - Vietnam
qtho@cse.hcmut.edu.vn
³University of Eastern Finland - Finland
jarkko.suhonen@uef.fi

ABSTRACT

Studying programming is complicated and time consuming, especially for beginners. According to our experience, learning programming techniques can be troublesome for many under-graduate engineering, computer science and information technology students. Previous research has shown promising results when using game-based solutions to support learning of programming concepts. The existing solutions have used mainly competitive contest and embed educational points to motivate students to learn programming. However, very few of the existing studies pay attention on programming for university students, especially in developing country context, including Vietnam. In this paper, we propose a Mistake Chasing Game model that encourage students to play and develop the game, aiming to increase their interest to learn mobile programming concepts. The game is developed on cross-platform mobile devices including smart phones, phablets, and tablets which are widely used among students. The preliminary evaluation results of the Mistake Chasing Game model indicate that students are willing to accept our approach and they reacted positively to the experience of playing and developing the game.

KEYWORDS


1 INTRODUCTION

Studying programming languages usually means to learn a new language talking with the computers. Quite often, vocabulary, syntax, and error messages of the language are difficult to remember and correctly addressed [1]. At under-graduate level, students are required to learn a specific programming language and learn to code applications, which sometime is not their favorite activity. Therefore, some students struggle with programming, which leads to inefficient programming languages teaching. We have faced the same difficulties with under-graduate students in Faculty of Information Technology (FIT) at Nong Lam University (NLU), Ho Chi Minh City, Vietnam, who are required to gain knowledge about mobile programming and learn how to create programs on mobile platforms. Thus, our intention in this work is to improve FIT students’ interest towards studying mobile programming. Instead of requiring student to spend their time reading programming books, we created a digital game based learning solution for students to play and develop educational mobile games. We have tested the game in real-life learning scenario at FIT in NLU, and the purpose of this paper is to report the first results of the experiment.

The rest of the article is structured as follows. In Section II, we give a background information related to e-learning and game-based learning for programming education. In Section III, we present the pedagogy approaches to create our game model. In Section IV, we present the Mistake Chasing Game. Section V explains the experimental result, which expresses our basic promising outcome. Finally, Section VI concludes the paper.

2 BACKGROUND: E-LEARNING AND GAME-BASED LEARNING FOR PROGRAMMING EDUCATION

¹ http://fit.hcmuaf.edu.vn/?lng=en
² http://en.hcmuaf.edu.vn/
Nowadays, social media, mobile programming, cloud computing and internet of things are the technologies to be applied in several fields, including computing education [6, 7]. Thus, various e-learning solutions have also been proposed for programming education, including use of mobile devices, social network sites, expert systems [1, 3, 4]. For example, social networking sites have been applied to help students to share source code and their learning experiences [18]. On the other hand, mobile learning platforms have been found to increase collaboration and support active participation of students during courses [15]. Concrete examples mobile learning solutions in computing education include developing interactive lectures for handling large classes [16], creating visual mobile learning objects for various topics [17], supporting peer to peer interaction [18] and peer code review [19], as well as providing tangible solutions by integrating robotics with mobile devices [20]. Most recently, also context-aware ubiquitous learning and internet-of-thing have been applied in computing education in order to create ubiquitous and interactive learning experiences [2, 9, 10].

**Digital game based learning** (DGBL) is the term used for game playing that has defined learning outcomes DGBL have been used in computing education, because games are attractive, novel, provide a better atmosphere and help keep the learner focused on the tasks [3, 6, 7]. Previous research has shown that DGBL approach can be used to support learning of various computing topics, such as artificial intelligence, database management, human-computer interaction, computer security, computer systems, algorithms and software engineering [5, 6, 21]. For example, Jordine et al. introduce a practical mobile gaming approach for learning Java programming, where students learn programming by creating mobile games [22].

However, we were able to identify only few DGBL-solutions for developing countries, especially on mobile devices. Moreover, none of the existing studies report about the concept of integrating the game players to join game development and improvement.

**3 PEDAGOGICAL APPROACH – EDUCATIONAL GAMES WITH PLAYING AND DEVELOPMENT**

In our experience, under-graduate students at FIT prefer playing games to studying books or going to the lectures. Vietnamese students are also shy to communicate, and they do not often discuss with lecturers and other students [8]. Information technology students in Vietnam generally have also challenges to describe the most complicated concepts and topics related to information technology, especially on logical thinking of the matters. On the other hand, the lectures are often not capable of identifying those topics that would require special attention during courses.

At the same time, we have noticed that learning mobile development and programming is becoming very important in Asia Pacific region. According to several reports (Garner3 and IDC4 reports) from information technology research and advisory companies, growth in smart phones market has increased quickly, especially in Asia Pacific region. Moreover, Vietnam and Thailand have strong market share and market growth in the recent years. As the obvious trend, mobile application development becomes a tremendous source of job demand for millions, particularly in Vietnam outsourcing industry. In order to become a mobile developer, a student need to receive appropriate education related to mobile development and programming. For example more than 60% of the student in FIT at Nong Lam University wish to study mobile development and want to become software mobile developers. At the same time, several courses in FIT’s academic program are supporting mobile development. In recent years, despite the growth in graduated software mobile engineers, that resource demand in information technology industry in Vietnam has not been satisfied.

![Figure 1: Updated paradigm shifts in e-Learning.](image-url)
In our work, we aim at improving education of mobile development and programming for undergraduate students in FIT in two aspects. First, we introduce a novel mobile learning game, called Mistake Chasing game. We suppose that teaching important programming skills via the game makes studying of programming more interesting. However, at the same time we want to integrate the development aspects into the mobile game experience. We call our model d-learning (educational games with playing and development), which is expansion of well known concepts of e-learning, m-learning and u-learning as seen in Figure 1.

As seen in Figure 1, e-Learning uses digital devices (usually computers and networks) to create a learning environment [10]. On the other hand, in the second stage, m-Learning enhances the mobility of learning locations, data storage, learning facilities by mobile devices (all hand-held devices, usually smart phones, tablets and laptops). In m-Learning, major technologies include mobile devices, wireless communication and online storage. In the third stage, the u-Learning comes from the features of context-aware ubiquitous learning, usually requires sensor and RFID (radio frequency identification) features of the devices.

After u-Learning, we propose another stage called d-Learning, requiring students not only to use e-Learning m-Learning and u-Learning solutions, but also to participate to the development of the solutions and learning environments. It is obvious that d-Learning is not an advanced stage of u-Learning due to its different approaches and fields of study, which requires development ability of the learners. But d-Learning is a following shift of m-Learning if the game is developed and played on a mobility platform. Beside the requirement of mobile devices and wireless communications, d-Learning needs a development framework including collaboration tools, cloud-based data storage, e-Survey and a development framework.

3 MISTAKE CHASING GAME – PLAYING AND DEVELOPMENT APPROACH TO LEARNING PROGRAMMING

In this section, we will introduce a DGBL-solution Mistake Chasing Game for under-graduate students in FIT using the d-Learning approach. The Mistake Chasing Game is developed by Mobile Development Team in FIT.

3.1 The Game Context

The context of the Mistake Chasing Games originated by the awareness that several students commit mistakes in Java programming. Training the student to recognize the errors can help to provide knowledge about the errors and quickly solve committed errors. The base for the mistake chasing created upon the List of Top Ten errors in Java programming [11] and the classification of programming errors [12, 13], as seen in Table 1.

<table>
<thead>
<tr>
<th>ID</th>
<th>Top Ten errors in Java Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Null pointers</td>
</tr>
<tr>
<td>2</td>
<td>Capitalization errors</td>
</tr>
<tr>
<td>3</td>
<td>Preventing concurrent access to shared variables by threads</td>
</tr>
<tr>
<td>4</td>
<td>Forgetting that Java is zero-indexed</td>
</tr>
<tr>
<td>5</td>
<td>Writing blank exception handlers</td>
</tr>
<tr>
<td>6</td>
<td>Confusion over passing by value, and passing by reference</td>
</tr>
<tr>
<td>7</td>
<td>Comparing two objects (== instead of .equals)</td>
</tr>
<tr>
<td>8</td>
<td>Comparison assignment (= rather than ==)</td>
</tr>
<tr>
<td>9</td>
<td>Mistyping the name of a method when overriding</td>
</tr>
<tr>
<td>10</td>
<td>Accessing non-static member variables from static methods (such as main)</td>
</tr>
</tbody>
</table>

The list of well-known errors listed in Table 1 are the ones the players needs to identify in the Mistake Chasing Game. In the game, the players locate and classify mistakes in source code as quick as possible. The more errors a player can find from the code, the higher score he/she can achieve from the game. Through error management, the players will improve their knowledge of reading source code, addressing errors and overcoming problems related to errors. Later on, the player will have opportunities to increase his/her scores in Single-Play and Chasing Modes of the game.

3.2 The Single-Play Mode

In the single-play mode, the player takes a time-limited game session alone. The game result is counted on number of correct answers and the complexity of questions that the player committed. The player’s score is updated depending on the player history and current achievement. The single-play mode helps the player to improve score, as long as the player intend to increase it by replaying the game. However, the improvement decreases and is upper-bounded when player repeats his/her playing in single- play mode several times.

Single-play mode relies heavily on creating fun to draw the player into the experience and to create a sense of investment. By asking player to guess the answers,
much information on programming language can be provided and progression base on the player history. It is also worth to mention that question complexity is updated as the game server calculate percentage of wrong commitment to a question. The higher the incorrect percentage committed previously, the higher the question complexity is updated. 

3.3 The Chasing Mode

In the Chasing Mode, up to four players can play simultaneously. The multi-player game option allow players with competition and rivalry, providing them with social communication absent from single-player games. In the Chasing Mode, players achieve a common goal in a time limited constraint (Figure 3). However, if a player can chase a correct answer, the question is locked for other players. The Multi-Player Mode requires players to play together over the internet. Score of the winners are updated regarding to their achievement and current scores of other players, by a predefined formula. As the result, score updates help to create fun and competitive approaches among several players. 

Beside the time-limitation, the Chasing Mode requires players to be the leader in selecting the correct answers, comparing with other players. If a player is success to commit a correct answer, that question is set to be disable to all other players. On the screen, the progression of all players in the game session is reported, showing their correct committed questions. The game session due happens if either time is over, or one player is success to gain correct submission of fifty percent (50%) of the playing questions. After the games, we alter score of players to create competition accordingly.

Figure 2 shows a screen shots of a playing session in the Single-Player Mode. In Figure 2a, the main screen shows a sample source code segment with errors. The player is required to choose correct answers by either detecting the errors or filling in the blanks. Remaining time is noticed on the right top of the screen. At any time, the player can re-check game guidance by pointing to the question mark, notifying Help, on the right top corner. Figure 2b shows screen shot of the page after the Help button is activated. All sessions that the players have gone are recorded in playing history.

Figure 3: Screen shot for Chasing (Multi-Players) Mode.

3.4 The System Implementation Overview

We used Java and Web-service technologies to implement the Mistake Chasing Game. Figure 4 shows the system overview of the related techniques. The players use smart mobile devices to play the game, which is hosted on a cloud. There are two cloud servers including cloud application server and cloud database server. In the application server, Apache Tomcat is the web server. We used HTML-5 and CSS-3 technologies in order to implement a web application for cross platforms purposes. JSF (Java Server Faces) and Prime Faces are the tools to develop graphic user interface on mobile web application. The other techniques enable the
application server to interact with database server. Most of these cloud services are light-weighted because we want the game to be played smoothly on cross-platform mobile device, with small memory and limited performance.

Figure 5: The Software Architect.

From another viewpoint, Figure 5 presents the Software Architecture of our application. The Mistake Chasing Game is implemented using three Layers: Database Layer, Core Layer and Application Layer. In the Database Layer, only services of database management is presented. On the other hand, the Core Layer provides the Application Programming Interfaces (APIs) to access Database Layer, supports Web Services and User Interfaces. The Core Layer work as the middle Layer to enable and authenticate accesses from Application Layer to the lower Database Layer. The top Application Layer provide interfaces for applications such as Score Report, Score Statistics Engine and Games Applications. The two engines in Figure 5 enables updating Game Bank and Question Bank. Question Update Engine imports and exports Question Bank from different sources of data, including contribution from lecturers and students to update Database. The above Layers create a development portal for cross-platform mobile web development for hand-held devices.

3.5 The Development Framework

Figure 6 presents the tools that are utilized in the development environment. Since we encourage all players (including lecturers and students) to collaborate developing the Mistake Chasing Game further on, we need to provide an easy development environment to share data and provide discussion.

The website Trello\(^5\) provides easy yet powerful space to conduct discussions on different topics of the game. Developers can share source code, documents and experience on the space. More instant methods such as university email and SMS can also be used freely. However, most of discussions can be easily hold on Trello. To call more collaboration among students, we encourage students taking related courses to join playing and developing the game, such as Mobile Programming, Network Programming and Projects on Computer Network & Telecommunications courses at FIT. These courses have Edmodo\(^6\) space to be used as a Content Management System, which is also a tool for collaboration space in our system. Our system in FIT is organized that all students need to check their Edmodo for up-to-date information among the lectures and other students. Therefore, it is a powerful tool to inform students with new updates. To help storing and managing versions, Github\(^7\) and Dropbox\(^8\) are used respectively. Finally, to archive feedbacks from players and developers, we use eSurveyspro\(^9\).

In Vietnam, developing cloud based applications has created high workload for the software industry and a giant demand in human resource production. In Nong Lam University, the students are also keen to learn and obtain new skills of the field, supporting their work after graduation. Investigation from Vietnamese industry shows that the interviewer usually ask for cloud computing and mobile development experience when they conduct the interviews with recruiting engineers.

---

\(^5\) Website: www.trello.com

\(^6\) Website: www.edmodo.com

\(^7\) Website: www.github.com

\(^8\) Website: www.dropbox.com

\(^9\) Website: www.esurveypro.com
Therefore, the collaboration space creates interesting learning demand among students and is helpful for their future career.

4 EXPERIMENT WITH THE MISTAKE CHASING GAME.

4.1 Context

In order to evaluate the Mistake Chasing game in real-life settings, we performed an experiment with undergraduate students at FIT, Nong Lam University, Vietnam. Also we wanted to commit the students to the future development of the game by collecting feedback, new ideas for game model and co-improve the current version of the game.

The game was introduced to learners in Mobile Programming Course (ID 217274). Also the game concept was applied in several other courses, including Computer Network and Telecommunication Projects (ID 214287), and Advanced Java (ID 214286). The students were required first to sign up an account in the home page of the game (http://nonglamit.net), using a smart mobile device. Later on, the player will need to activate his account by email and login for his/her first playing session. History, score, knowledge improvement of a player are recorded and reported to him/her each time the player enters to the game.

4.1 Questionnaire about the game

In order collect feedback, we created an online questionnaire consisting of ten questionnaire items (Table 2). Each question was formulated as a statement, which the players needed to answer using a 6-point Likert Scale (strongly disagree, moderately disagree, slightly agree, moderately agree, strongly agree and no opinion). We asked player, for instance, to evaluate his/her each time he/she enters the game.

We used an online survey tool eSurveys to collect the feedback. Altogether around 170 players has tried the game and 56 players answered the questionnaire, which means that response rate was about 32.9%. The distribution of answers to ten questionnaire items is presented in Table 2, while Table 3 includes the total number of answers to options: 1=strongly disagree, 2=moderately disagree, 3=slightly agree, 4=moderately agree, 5=strongly agree (e.g. excluding those who selected “no answer” option), as well as the mean, median and mode values of the answers.

When looking at the results, we can observe that the players strongly agreed that the structure of the game was clear (average score of 4.27). Also the players strongly agreed that Mistake Chasing game supported self-reflection. On the other hand, questions related to the players’ understanding of the game and sharing of ideas received lower ratings than other question items. We can see the in general level the player’s had a good experience with the game, and they could see the educational value of the game. However, the low ratings related to the understanding of the game and sharing of ideas require attention when the game is developed further on.

Table 2: Distributions of answers to the questionnaire (N=56).

<table>
<thead>
<tr>
<th>Questionnaire item</th>
<th>Strongly disagree</th>
<th>Moderately disagree</th>
<th>Slightly agree</th>
<th>Moderately agree</th>
<th>Strongly agree</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned something new from the game</td>
<td>8.9</td>
<td>5.3</td>
<td>25</td>
<td>16.0</td>
<td>30.3</td>
<td>14.2</td>
</tr>
<tr>
<td>Game easy to understand</td>
<td>17.8</td>
<td><strong>48.2</strong></td>
<td>7</td>
<td>5.3</td>
<td>17.8</td>
<td>3.5</td>
</tr>
<tr>
<td>Participation in online environment</td>
<td>3.9</td>
<td>17.6</td>
<td>3.9</td>
<td>49</td>
<td><strong>19.6</strong></td>
<td>5.8</td>
</tr>
<tr>
<td>Learning environment facilitated networking</td>
<td>5.3</td>
<td>8.9</td>
<td>42.9</td>
<td>12.5</td>
<td><strong>21.4</strong></td>
<td>8.9</td>
</tr>
<tr>
<td>Sharing ideas in online learning environment</td>
<td>28.5</td>
<td>25</td>
<td>7.1</td>
<td>17.8</td>
<td>21.4</td>
<td>0</td>
</tr>
<tr>
<td>Mistake Chasing game met its purpose</td>
<td>8.9</td>
<td>5.3</td>
<td>12.5</td>
<td><strong>53.5</strong></td>
<td>16.1</td>
<td>3.5</td>
</tr>
<tr>
<td>Mistake Chasing game met learners’ expectations</td>
<td>7.1</td>
<td>23.2</td>
<td>7.1</td>
<td><strong>33.9</strong></td>
<td>21.4</td>
<td>7.1</td>
</tr>
<tr>
<td>Mistake Chasing game supported self-reflection</td>
<td>7.1</td>
<td>12.5</td>
<td>7.1</td>
<td><strong>39.2</strong></td>
<td>25</td>
<td>8.9</td>
</tr>
<tr>
<td>The structure of the game was clear</td>
<td>0</td>
<td>5.4</td>
<td>17.8</td>
<td>16.1</td>
<td><strong>53.6</strong></td>
<td>7.1</td>
</tr>
<tr>
<td>Technical problems experiences when using the game</td>
<td>10.7</td>
<td>17.8</td>
<td>12.5</td>
<td><strong>39.3</strong></td>
<td>16.1</td>
<td>3.6</td>
</tr>
</tbody>
</table>

Also the questionnaire included open questions, where the players were asked to express their opinions and give feedback about the Mistake Chasing Game. The players were asked to share what kind of new ideas the players gained from the game, possible amusement factors of...
the game, learning aspects, opinions about the development framework. Also the players were able to give feedback about the current implementation of the game. In Table 4, we have collected the most interesting feedback revived from the open questions.

Table 3: Results of the questionnaire

<table>
<thead>
<tr>
<th>Questionnaire item</th>
<th>Number of answers with opinion</th>
<th>Mean</th>
<th>Median</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned something new from the game</td>
<td>48</td>
<td>3.63</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Game easy to understand</td>
<td>54</td>
<td>2.56</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Participation in online environment</td>
<td>48</td>
<td>3.67</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Learning environment facilitated networking</td>
<td>51</td>
<td>3.39</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Sharing ideas in online learning environment</td>
<td>56</td>
<td>2.79</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Mistake Chasing game met its purpose</td>
<td>54</td>
<td>3.65</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Mistake Chasing game met learners’ expectations</td>
<td>52</td>
<td>3.42</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Mistake Chasing game supported self-reflection</td>
<td>51</td>
<td>3.69</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>The structure of the game was clear</td>
<td>52</td>
<td>4.27</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Technical problems experiences when using the game</td>
<td>54</td>
<td>3.33</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4: Feedback on open questions.

<table>
<thead>
<tr>
<th>Topics</th>
<th>Open discussion feedbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>New ideas</td>
<td>Got new ideas of educational games on mobile</td>
</tr>
<tr>
<td></td>
<td>Create players hope to share their updates on games with friends</td>
</tr>
<tr>
<td></td>
<td>Create players wish to find where is the game</td>
</tr>
<tr>
<td>Fun</td>
<td>Much fun is gain winning a friend</td>
</tr>
<tr>
<td></td>
<td>Very interesting if new game ideas is applied</td>
</tr>
<tr>
<td>Learning</td>
<td>Learn new programming techniques from game</td>
</tr>
<tr>
<td></td>
<td>A very interesting way of learning, especially for programming</td>
</tr>
<tr>
<td></td>
<td>Several important technologies are learned when joining the learning model</td>
</tr>
<tr>
<td>Development</td>
<td>Difficult but interesting development and collaboration framework</td>
</tr>
<tr>
<td></td>
<td>Take long time to be familiar with development environment. Better be applied in more than two semesters</td>
</tr>
<tr>
<td>Feedback</td>
<td>Can feedback on new questions, games return faster?</td>
</tr>
<tr>
<td></td>
<td>Can feedback go on SMS?</td>
</tr>
<tr>
<td></td>
<td>Can open additional games scope to other subjects, information technologies related topic?</td>
</tr>
</tbody>
</table>

The collected feedback indicates that the response from the players is encouraging and promising. For instance, the player reported that they have learned new programming techniques from the game. On the other hand, the players gave ideas for future development of the game and development framework. Based on the first experience, we are in good condition both to extent the game, as well as follow the several interesting proposals given by the players.

4.2 Game Development Verification

The second part of our experiment is game proposals verification and deployment. We received several suggestions on how the Mistake Chasing Game could be further improved, including new questions, new game modes and other type of improvements to enhance the current version of the game. To verify the suitability of proposals, we created a peer review process. Each proposal is reviewed by three other hidden players before being taken into account. A player can see feedbacks on his/her proposal if all the works assigned to him/her are finished. We have enhanced our Question Bank to 150 questions of different types in the Mistake Chasing Game.

Table 5: Ideas for development.

<table>
<thead>
<tr>
<th>ID</th>
<th>Ideas for development</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mistake identification</td>
</tr>
<tr>
<td>2</td>
<td>Filling in the blank with missing code</td>
</tr>
<tr>
<td>3</td>
<td>Locating the error positions in source code</td>
</tr>
<tr>
<td>4</td>
<td>Multi-player mode with update current status to other players using Signal-R</td>
</tr>
<tr>
<td>5</td>
<td>Multi-player mode using notification services</td>
</tr>
</tbody>
</table>

Table 5 summarizes the game proposals provided by players. Some of the proposals have been selected to be implemented, utilizing development from players. The learning model help to receive more proposals from players for later improvement.

5 CONCLUSION

This study demonstrated the possibility of using computer game framework to encourage students to play and develop programming games for learning programming. The d-Learning model creates encouragement for both teachers and students in undergraduate programming courses. We perform experiment with students in FIT, NLU Vietnam. Currently, new sign-up and update game playing information are being
sent to our game servers. New discussion and game improvement messages are also sent everyday.

The proposed d-Learning model enables game-based programming skills, development framework and collaboration processes. Students also can learn much on current hot topics include cloud computing, Java mobile applications and cross-platform mobile programming. The response of the players towards the game was mainly positive, and the players were able to give interesting insights on how to develop the Mistake Chasing Game and d-Learning model further on. Thus, we aim to enhance and expand the model to a game portal where players can develop their own educational games, and get connected to the learning environment. We also plan to develop new games for other areas related to programming and mobile development.

ACKNOWLEDGMENT

This work is supported by the North South South Garage Computing Project, funded by CIMO [14].

REFERENCES