The Development of Automatic Image Marking System towards Better Assessment Monitoring for Computer Science Programming Students

Aeman I.G. Masbah#1**, Joan Lu***2**, Qiang Xu**#3

#School of Computing and Engineering, University of Huddersfield
Huddersfield, UK
[1]Aeman.Masbah@hud.ac.uk, [3]q.xu2@hud.ac.uk
*School of Computing and Engineering, University of Huddersfield
Huddersfield, UK
[2]J.lu@hud.ac.uk

Abstract

The implementation of technology to support both instructor and student enables both to perform better in the educational context. Hence, automatic markers can be an improvement over human markers because their results are reliable and repeatable. This paper presents a framework of an automated marking application (iMarking®) assisting both instructors and students through the execution of java language system towards better assessment monitoring for computer science programming students. The prototype of the proposed system has been evaluated and result was documented as a recommendation for future researches.

Keywords: *automated assessment monitoring, computer science students, (imarking®), optical character recognition, operational test and evaluation*

1. Introduction

The notion of education management refers to organizing and handling educational resources in such a way that the mission is accomplished within well-defined range, value, time and cost boundaries [1]. In respect to this statement, handling students' academic tasks and records play a major role in the development and achievement of the goals of educational Landscape. Hence, the implementation of technology to support both instructor and student enables both to perform better in the educational context.

Particularly, computers are no longer used basically as techniques for precise tasks and with the spread of computer technology along with the transformation of educational environment both have raised the inspiration towards more investigations into more helpful tools and automation of routine responsibilities [2].

More particularly, Assessment is a vital component of academic tasks. Scholars [3] and [4] claimed that assessment can have a solid outcome on student learning progress. Students learn best by frequent assessment with rapid feedback. Unfortunately, assessment can be a difficult mission for instructors. It takes time both to generate the assessments and to mark them. Automatic markers can decrease time instructors devote marking assessments. Manual marking can only be prepared by knowledgeable programmers. With great class sizes it is exclusive for lecturers to physically e[amine each and every student's result. The procedure needs the engagement of more instruction supporters, who costs less than lecturers, but is more variable in the quality of their result review [5].

In contrast, automatic markers can be an improvement over human markers because their results are reliable and repeatable. Thus, automatic markers outcomes will be the similar without regard to the order in which the answers are presented, and they are able to return results much faster than humans.

This work is an extension of the previous work [6] where the focus was only in the adaption of OCR service for automatic image marking, the main contribution of this paper that is proposed a novel evaluation metric to calculate the final mark of the submitted student answers. The final mark is based on calculating the percentage of matching and mismatching using some new calculation equations.

2. Arguments in Favor of Using an Automated Assessment System

Software specialists have always encouraged education insinuations to demonstrate more software engineering towards better assessment monitoring. For instance, [7] stated that Computer-based Assessment System (CAS) has become like a backbone of computer based education industry. Likewise, lecturers and education decision-makers are always overburdened with workload. An automatic marking system can decrease this overhead and helps lecturers to focus on other important issues of designing program tasks. According to the indications emerged from [7] the role of automated marking system can be summarized as it delivers additional aids in terms of reliability, thoroughness and effectiveness. One of the major advantages of such systems is that timely feedback can be given to students on their performance accordingly

3. Central Theme of The Paper

Nowadays, the capability to program is gradually important, with application in a comprehensive collection of careers. Accordingly, attention among college learners in learning how to program has also increased, and more individuals are trying to acquire programming on their own in order to gain an edge in their careers. Becoming a worthy computer programmer needs both practice and response.

The degree at which a beginner programmer brands development be determined by greatly on how much informative feedback the programmer obtains from experienced lecturers. However, capable computer programmers who can deliver such feedback are a limited resource. Hence, this paper presents a proposed development of an automated marking application (iMarking®) which hopefully would assist both instructors and students through the execution of both java and C++ languages towards better assessment monitoring for computer science programming students.

4. Related Work

A. Automated System Throughout Literature

Automated assessment takes place in an assessment platform (e.g. [8, 9 & 10]). Most modern assessment platforms are Web based. Students use them with their browsers and submit programs to the server where the programs are typically tested. Many of the platforms include course management features taking care of submissions, finding the best submission and collecting results for the teacher.

B. Recent Automated Assessment Tools

Amelung et al. [11] executed a program that splits all concerns of handling learners, projects, and submissions from the authentic analysis. The program contains two measures. The frontend manages storing of tasks and answers, appropriate management of submission times and re- submissions, communication of outcomes to learners, ranking of the grades, and indicators for individual learners. The given responses is performed and the output of a learner answer can be matched to that of a prototypical solution for a set of assessment documents, or the task can be verified for properties which must be satisfied by precise programs.

Web-CAT [12] is a structure for programmed assessment. The program requires learners to pass their own exams beside their answers. The lecturer has an executed a model resolution on the server side. Feedback and comment are produced upon the execution of learner answers on student exams. As a final move, the programmed model is linked with statement as well as branch information arriving at a solid measure of the range of the performed test.

Naud'e et al. [13] and Wang et al. [14] together used diagram similarity to evaluate learner programs. In this diagram, Learner are first examined and shifted into system requirement graphs. The task is completed by a graph similarity quantity in contradiction of a pool of prototypical scheme requirement graphs. Furthermore, Easyaccept [15] created scripting system to describe assessments and estimated results. Thus, Easyaccept delivers an easier technique for lecturers to mark assessments for learners.

Hollingworth's grader is one of earliest application in automated program course. It was designed to evaluate card programs [16]. Presently, there are a number of other applications designed to function in a similar way including the Online Judge [17], and the most recent one Sakai. Sakai, which was designed by Suleman was more advanced [18].

Although the fact that Automated Marker differs in terms of names and peripheries, they still have the same principal function of evaluating programs codified by learners indirectly through the performance. The indirect evaluation process has been accompanied with a number of limitations. As noted by researchers, indirect evaluation is limited within certain quality of feedback, largely hinders evaluation and is over sensitive even to minor mistakes.

In addition, it cannot mark interactive programs, non- textual programs, and tasks that have specific algorithms. Specific algorithms include animations and other drawing programs. Pragmatically, researchers should explore on ways to upgrading functionalities of existing automated markers and recommend on how current limitations should be curbed.

File-system-based organizational strategies are also widely used approaches in automated program makers as documented in literature. For instance, Isong marker [19] was designed to compile learners program automatically. In this case, there is comparison between data provided by an instructor against the learner program performance. Isong marker was designed using Unix-shell scripts.

But then before Isong marker was designed, Reek marker, which was inexistence even after Isong Maker had been designed, served similar functions [20]. For instance, Reek marker Unix-based system designed for inductor courses of programming. In addition, Reek marker also used file-system organisational strategy to in evaluation of assignments and performance.

Performances are marked against the data provided by an instructor. Therefore, an instructor manages markers feedback and the process of evaluation. Nevertheless, there is also another marker referred to as BOSS. BOSS is designed with a battery of Unix-based programs that uses file-system- based organizational strategy for submission of evaluation tests that is tested against instructor-provided test criteria [21].In these studies, the system that has been proposed will vary from other systems based on efficiency and accuracy in automatic marking operations. It will consider the use of new technology that will be based on OCR or images.

5. Evalution of Drawback of Earlier Efforts

Kay et al. [22] indicated that time has come to devote effort and resources to develop flexible tools for automatic program evaluation. Hence, we won't be able to assess students work validity and reliability by manual means. Understandably given the concern about the cost of marker and escalating demand of test recruitments, the following are the critical shortcomings in earlier efforts regarding automated marking systems:

1) Student may browse some practices that are not reliably as a part of design.
2) The value of feedback that students obtain should be directed towards continual development. In this case, students must obtain concrete phases to develop better understanding and.
3) The automated marking software must be thoroughly implemented through the curriculum in a manner that makes it an ingrained part of all evaluation process.

6. The New Approach Towards Automated Marking Process

To overcome a lack of research in this area and fill this gap, the proposed design has been seen as the proliferation of techniques by using Optimal Character Recognition (OCR) by extracting relevant coding feature from assignments' image. This entire extracting image process has let to content based image retrieval system (e.g. [23, 24]), have been enormously facilitated to make it possible to index large collection of images of coding data provided by the students and then process hitherto required automatic effort.

Despite the fact that OCR facilitated in such advance manner there are still many challenge to develop iMarking® in trying to deploy more user-oriented system. The reason for that challenge is actually the standard output of images provided by the students continues to consist of low features, such as color textures and quality. It is a real time examination system that provides an environment for the compilation and execution of designed prototype and the automatically mark the student's assignments as per provided results criteria by the examiner.

The prototype has been designed based on Web interface that enables student to submit their assignments in the shape of image and then get marked automatically and also get feedback accordingly. It is hoped that by sharing the web based interface as a part of coding data can be constructed to examine a range of programming problems. However, there is significant and practical pedagogical value in computer based automatic assessments system of such tests. The task of soring he image coding results and then comparing with the pre-existed answers in an accurate level is difficult and relative unbearable. There is also a challenge face that the quality of images was also identified bit low and there was intense need to test later that how to improve the quality and the reading of image for scoring the content portion for large scale assessments.

7. Imarking® Architecture

iMarking® is a combination of optical character recognition (OCR), web technology, and database. This section consists of three sub-sections: (1) Login, (2) submission process and (3) marking process. Fig.1 shows the architecture of iMarking®.

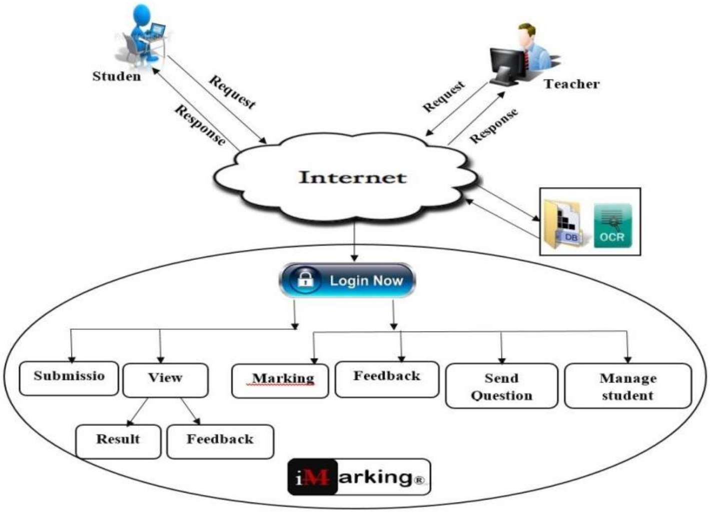
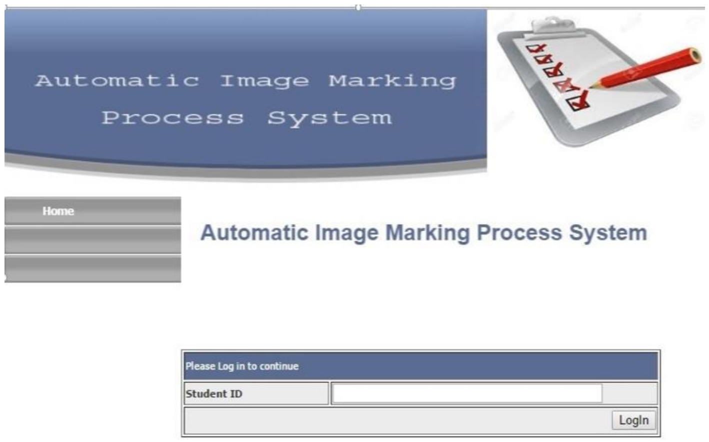


Fig.1 Architecture of iMarking® system

iMarking® system consists of three main phases, users interface which designed by Java language, image processing by OCR, and database.

A. Login

The system is used to conduct user (Students & teachers) authentication and anchors login process. This system stores the list of users created for the system and the users can provide their password for successful logins.


Automatic Image Marking Process System

Automatic Image Marking Process System

Please Log in to continue

Student ID

Fig. 2 Shows us student log-in page

B. Submission Process

After the student logged into iMarking® by ID number, the system will appear following page to the students:

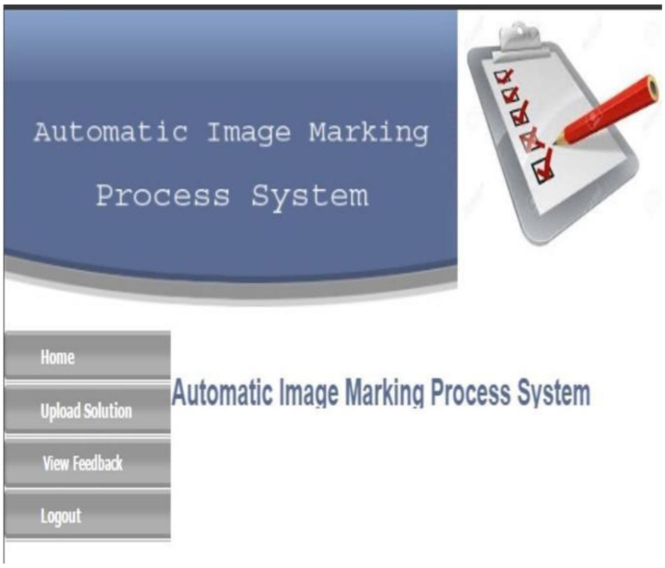


Fig. 3 Shows us upload assignment

The iMarking® system provides students interface to upload their assignment answers,get feedback and view thire result.

C. Marking Process

After the teacher logged into the system by using his/her ID number, the following page will appear to the teacher:

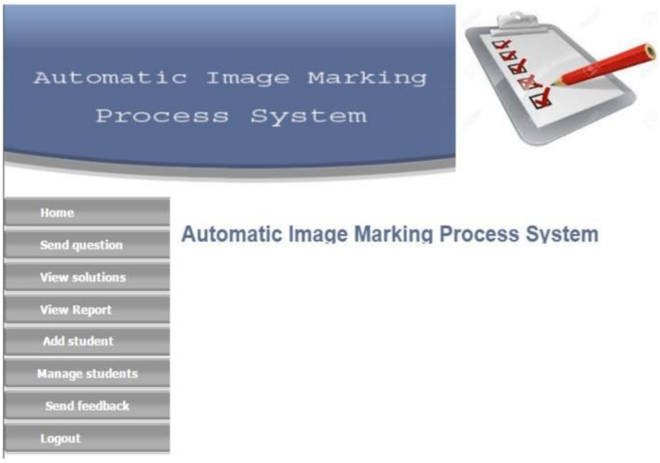
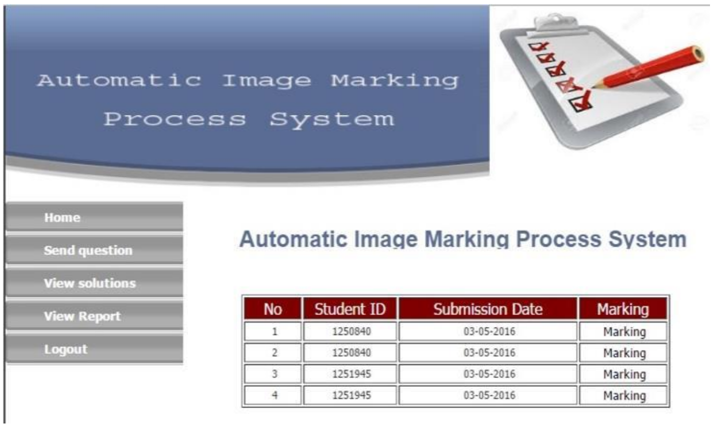


Fig. 4 The main page for the teacher

The marking process is described as follows:

- The teacher presses on view solution button, and then selects one of the submitted answers. Fig. 5 shows view solution page:

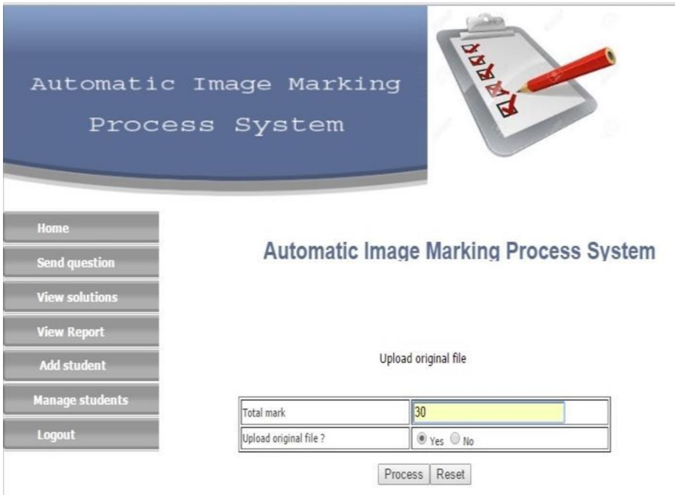


Automatic Image Marking Process System

No	Student ID	Submission Date	Marking
1	1250840	03-05-2016	Marking
2	1250840	03-05-2016	Marking
3	1251945	03-05-2016	Marking
4	1251945	03-05-2016	Marking

Fig. 5 View solution page

iMarking® allows to the teacher to upload the optimal Answer and to enter the assignment mark. Fig. 6 shows the upload of the optimal answer.



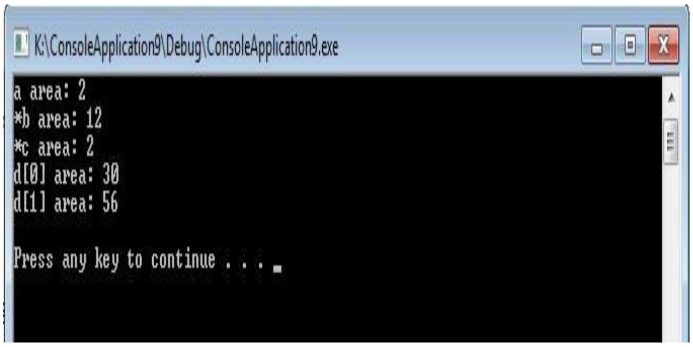
Automatic Image Marking Process System

Upload original file

Total mark	30
Upload original file ?	● Yes ○ No

Fig. 6 Marking submitted answer page

- iMarking® uses the OCR web service to extract the text fromthe submitted answer and the optimal answer. The system compares both texts (i.e., the submitted and optimal answer) and computes the final mark.



```
a area: 2
*b area: 12
*c area: 2
d[0] area: 30
d[1] area: 56

Press any key to continue . . .
```

Fig. 7 Submitted image that represents the student's answer percentage of matching

8. Experimental Result and Evalution

It is the author's belief that software measurement can indeed help in understanding, controlling and managing the development of a software product. The work in this paper has focused with automated assessment of students programming given tasks by instructors. Thus, assessing software quality was one major interest in this work. Therefore, the quality of our proposed system is reviewed thoroughly. Initially, the offered system was tested on a sample of 100 original teacher answers and compares the original file with student provided answer. Students were grouped and asked to upload their group answers on the programming questions that they were assigned. A specific optimal answer is then selected by the teacher and uploaded on the system and assigns marks to the questions. The system then uses a mathematical program to calculate the percentage for the marks using equation 1.

Percentage of matching =

$$\frac{Number\ of\ matched\ lines}{Total\ number\ of\ the\ optipal\ answer\ line} \ldots (1)$$

Also the system calculates the grade for each line in target file using equation (2)

Grade for each line =

$$\frac{Final\ mark\ for\ the\ question}{Total\ number\ of\ line\ in\ original\ file} \ldots (2)$$

The system calculates the final mark using (3)

$$final\ mark = mark\ of\ line1 + mark\ of\ line2 \ldots\ldots \\ + mark\ of\ linen \ldots\ (3)$$

In statistical terms, the optimal solutions are simultaneously represented in the programming assignment. The solutions are represented in a semantic space with the answers in a sequence in the system. A vector representation is used to classify the right representation of optimal answers. The answer which is close to the optimal answer is then selected by the vector. The final mark for the student equivalent to 30 for each student is shown in Fig. 8. At the end of the process, the submission and evaluation of the full sample is done to achieve the final mark and percentage for the groups.The final mark is achieved when the optimal answer and that which is submitted equal each other. A sample table is available in Appendix.

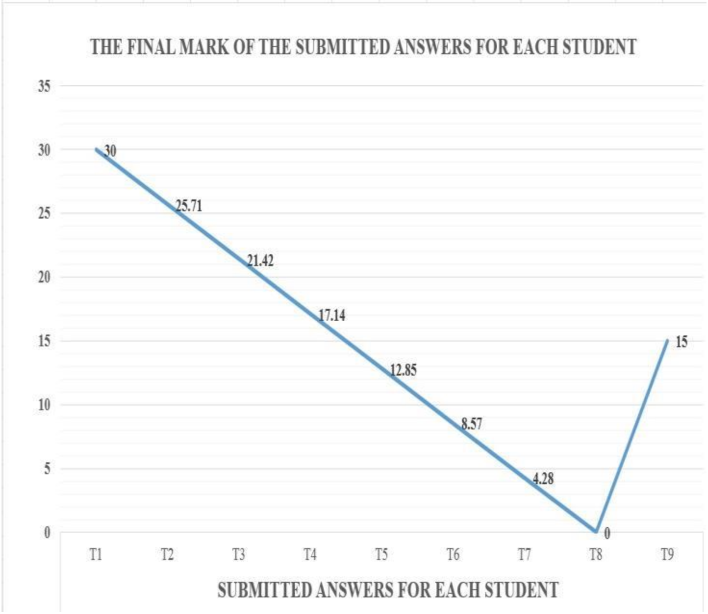


Fig. 8 The Final mark of the submitted answers for each student

Besides, the students then apply extensive examination and implement OCR to the given answers to achieve the given levels of accuracy. However, sometimes the quality of the image that is uploaded negatively affects the accuracy. For this reason, it was recommend that 200 DPI to 400 DPI resolution be used by students to make recognition better. This was followed by the extraction of the image content. After the extraction, the line of each of the content was compared with the corresponding line in the optimal answer. This was to check how close the two images were matching. In conclusion, the efficiency of the proposed system in automatically marking the student assignment is corroborated by the results of the experiment to be valid.

9. Conclusion and Future Work

The adoption of technology to support education process has been proven to be significantly valued and regarded by both students and instructors. Accordingly, the development of this automatic image marking system as a tool to automate the marking process for better assessment monitoring for computer science programming students has liberated teaching monitor to do a more creative effort. Furthermore, the presentation of iMarking® gives an alternate support towards securing the test environment. However, like any good story the possibility of a sequel has been built in and a further edition would be a possibility.

Future work, one gap is still faces the proposed system its syntax similarity, the future work will be how to solve the syntax similarity with our proposed system.

Refernces

[1]. D. Laurillard, Teaching as a design science: Building pedagogical patterns for learning and technology: Routledge, 2013.

[2]. A. Weinberger, H. Dreher, M. Al-Smadi, and C. Guetl, "Analytical assessment rubrics to facilitate semi-automated Essay grading and feedback provision," in Australian Technology Network Assessment Conference 2011: Meeting the Challenges, Perth, Western Australia, 2011.

[3]. D. Fowles, "Does marking images of essays on screen retain marker confidence and reliability," in International Association for Educational Assessment Annual Conference, 2008, pp. 7-12.

[4]. E. L. Jones, "An experiential approach to incorporating software testing into the computer science curriculum," in Frontiers in Education Conference, 2001. 31st Annual, 2001, pp. F3D-7-F3D-11 vol. 2.

[5]. D. Jackson and M. Usher, "Grading student programs using ASSYST," in ACM SIGCSE Bulletin, 1997, pp. 335-339.

[6]. A. Masbah and J. Lu, "Automatic Image Marking Process," presented at the INFOCOMP 2016, The Sixth International Conference on Advanced Communications and Computation, Valencia, Spain, 2016.

[7]. M. Y. Ozden, "Students' perceptions of online assessment: A case study," International Journal of E-Learning & Distance Education, vol. 19, pp. 77-92, 2005.

[8]. Stephen H. Edwards. Rethinking computer science education from a test-first perspective. In OOPSLA ’03: Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pages 148–155. ACM, New York, NY, USA, 2003.

[9]. Mike Joy, Nathan Griffiths, and Russell Boyatt. The BOSS online submission and assessment system. Journal on Educational Resources in Computing (JERIC), 5:Article 2 (28 pages), 2005.

[10]. Mike Joy and Michael Luck. Effective electronic marking for on-line assessment. SIGCSE Bull., 30:134–138, 1998.

[11]. M. Amelung, P. Forbrig, and D. R¨osner, “Towards generic and flexible web services for e-assessment,” SIGCSE Bull., vol. 40, no. 3, pp. 219–224, June 2008.

[12]. A. Shah, “Web-cat: A web-based center for automated testing,” Ph.D. dissertation, Virginia Tech, Citeseer, 2003.

[13]. K. A. Naud´e, J. H. Greyling, and D. Vogts, “Marking student programs using graph similarity,” Computers & Education, vol. 54, no. 2, pp. 545– 561, 2010.

[14]. T. Wang, X. Su, Y. Wang, and P. Ma, “Semantic similarity-based grading of student programs,” Information and Software Technology, vol. 49, no. 2, pp. 99–107, 2007.

[15]. J. P. Sauv´e and O. L. Abath Neto, “Teaching software development with atdd and easyaccept,” SIGCSE Bull., vol. 40, no. 1, pp. 542–546, Mar. 2008.

[16]. J. Hollingsworth, "Automatic graders for programming classes," Communications of the ACM, vol. 3, pp. 528-529, 1960.

[17]. B. Cheang, A. Kurnia, A. Lim, and W.-C. Oon, "On automated grading of programming assignments in an academic institution," Computers & Education, vol. 41, pp. 121-131, 2003.

[18]. H. Suleman, "Automatic marking with Sakai," in Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology, 2008, pp. 229-236.

[19]. J. Isong, "Developing an automated program checker," Journal of Computing Sciences in Colleges, 2001, pp. 218-224.

[20]. M. Luck and M. Joy, "A secure on-line submission system," Software-Practice and Experience, vol. 29, pp. 721-40, 1999.

[21]. K. T. Weber and J. S. Janicki, "Cardiopulmonary exercise testing for evaluation of chronic cardiac failure," The American Journal of Cardiology, vol. 55, pp. A22-A31, 1985

[22]. Kay D., Scott T., Isaacson P., Reek K., (1994), Automated grading assistance for student programs, Selected papers of the 25th annual SIGCSE Symposium on Computer Science Education, Phoenix, AR USA, March 10–12, 1994, Pages 381–382.

[23]. C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, et al., "QBIC project: querying images by content, using color, texture, and shape," in IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology, 1993, pp. 173-187.

[24]. H. Parkin and L. Thorpe, "Exploring student experiences of e-learning: A phenomenographic approach," in Bera Annual Conference (September 2-5, 2009). Manchester, UK: University of Manchester, 200