

# Plagiarism in e-learning systems: Identifying and solving the problem for practical assignments

Emil Marais  
Academy for IT, University  
of Johannesburg  
emar@rau.ac.za

Ursula Minnaar  
Academy for IT, University  
of Johannesburg  
um@rau.ac.za

David Argles  
School of Electronics and  
Computer Science,  
University of Southampton  
da@ecs.soton.ac.uk

## Abstract

*A big part of life long learning is the move from residential lectures to distance education. Distance education falls under the multi-modal policy of the teaching institution and thereby a change in student contact. The lecturer facilitating the distance education course is also faced with a problem where the quality and originality of submitted assignments need to be checked. This has always been a difficult task, as going through practical assignments and looking for similarities is a tedious job. Software checkers are available, but as yet, have not been integrated into popular online e-learning systems. If closer contact and warning to students are given at an early stage the problem is minimized as they know they are being closely monitored. As will be shown in this article, plagiarism is a current problem with online practical submissions. We will also show how this problem can be minimized through the integration of plagiarism checking tools and other checking methods into e-learning systems.*

## 1. Introduction

E-Learning systems ease the course administration burden of presenting a course and provide tools to present the information in an orderly and clear way. Although commercial and experimental systems have vastly improved over the last 5 years and have become learning portals they still lack certain tools. We will start by looking at the results obtained when allowing learners to remotely submit their assignments without any control of where and when they can submit. The only restriction was a cut-off date that had to be adhered to. Thereafter methods to limit the corrupt use of e-learning systems will be discussed and lastly the reasons for presenting all these methods in an easily manageable interface will be discussed.

## 2. The extent of cheating in e-learning systems

The statistics presented here is for second year B.Sc. Computer Science students and distance education students, taking a Java programming course that focuses on the practical application of the theory and practical knowledge covered in the course. This is not the first time that the students are exposed to programming as their first year covers problem solving in VB.Net and C++.

Each week a practical assignment is given that tests the student's knowledge of the week's work covered. The practicals and study guide state very clearly that this assignment should be completed individually. The definition of both "individual assignments" and "plagiarism" is also given. The students are then given a week to complete the assignment, and they may make use of the lecture notes, the Internet, book sources, university employed assistants or the lecturer if they have any problems.

The practical assignments count about 10% towards the students semester mark. Although this is not much, the knowledge needed to complete the practical assignments is tested again during written semester tests that count +40%. The reason behind this discrepancy in mark allocation is that the extent of corruption limits the amount that unsupervised assignments can count.

At the cut-off date/time the practicals are submitted and marked from the server. Therefore they can submit their assignment at any stage and from any computer with Internet access. The feedback facility of the e-learning system was used to give students feedback on the mark they received and comments on possible improvements to their submissions. Due to the large class size (193 students) and time limitations it is not possible to compare each practical with all others without the aid of a program.

Before we get to the details of how widespread the problem is, we first need to identify the reason why two similar practical submissions could be received from two students. The following is a list of possible reasons for the sources of the corruption:

- The theft of practicals due to a weak password. [1]
- Collaboration between students.
- Electronic corruption where a student breaks into the electronic submission area of a student or plagiarizes a practical form another source like the Internet.
- Lastly it should also not be allowed that a student has his/her assignment submitted by another student.

The next section gives statistics that shows how widespread corruption is in practical e-learning submission systems.

## 2.1 Statistics on the extent of corruption.

What are presented here are the results of a program that was written by the authors. The exact working of the program is not essential to the discussion, but the basic methods used were:

- Pattern matching based on a sliding window mechanism.
- Stripping of comments, variable names and formatting.
- Stripping of common code such as code given in class and common assignment code such as the creation of a TCP/IP socket in Java.
- Many more advanced techniques exist that is not covered in this article. [2, 3]

Post checking of assignments was done after each submission so that the student would have the result before the next practical submission. [3] This was done to allow the students to amend their corrupt behaviour. The sample group was 193 students with 5 practical assignments. Only active students (i.e. students that submitted 2 or more practicals during the semester) were included in the sample.

Figure 1 shows the amount of copiers in each of the 5 practical assignments while Figure 2 shows what percentage of submitters from the class of 193 students have copied. As can be seen from the above results, cheating is unacceptably high with electronic submissions. Therefore even with the simple sliding window technique used to check for copies those students that did not deserve their marks were caught.

Providing such a tool to the lecturer would obviously be beneficial.

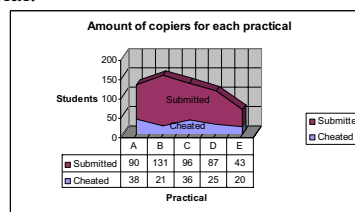


Figure 1: Amount of plagiarism for each practical

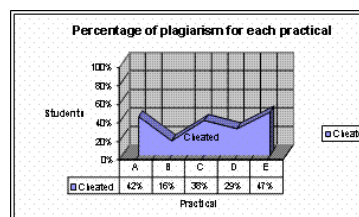


Figure 2: Percentage of plagiarism in each practical

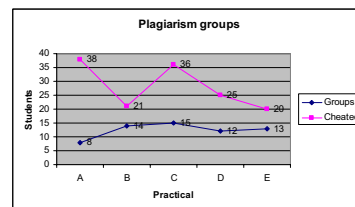


Figure 3: Amount of plagiarism groups

Figure 3 shows the amount of plagiarism groups. Each plagiarism group contains a uniquely identifiable solution from the plagiarised total. For example, of the 38 copied practicals that were found for assignment A, there were only 8 distinct solutions, which indicates that more than one student copied the same assignment.

This implies that there is a relatively small number of “original authors” who are prepared to share their work with others (and risk getting a 0 mark), which means that identifying these students could minimize corrupt use. Unfortunately identifying the original authors is extremely difficult and time consuming, and the penalty is not always big enough to discourage them. Making the penalty for copying higher than just receiving 0 is also an option but then you sit with 38 students that complain about the lecturer at the dean or higher authority. If a case is taken to such a high authority it normally takes months to resolve if either party does not back down. This is just one more reason for integrating such a checking method into the

e-learning system to thereby standardise the checking method so that it would not come into question every time a student complains.

The next section will show how pre-submission checking can minimize or even eliminate corruption when an assignments submission environment is controlled. Post checking can also be applied but the cheating can be reduced by applying these measures.

### 3. Monitoring the submission environment

The checking referred to here is for the programming code created by students for practical assignments and assessments. For written assignments many techniques and services are available to check for plagiarism. That does not work directly for programming assignments. A short list of available services and research is: [4, 5]

- CopyCatch: [www.copycatchgold.com](http://www.copycatchgold.com)
- TurnItIn: [www.turnitin.com](http://www.turnitin.com)
- MyDropBox: [www.mydropbox.com](http://www.mydropbox.com)
- Eve: [www.canexus.com](http://www.canexus.com)
- Plagiarism.com: [www.plagiarism.com](http://www.plagiarism.com)
- Jplag: [www.jplag.de](http://www.jplag.de)
- Copyscape: [www.copyscape.com](http://www.copyscape.com)

The checking of practical assignments/assessments can be done in one or both of the following ways:

- Pre-submission checking.
- Post-submission checking.

With pre-submission checking the submission process is constantly monitored for corrupt behaviour. If such behaviour is detected an alert can be raised.

With post-submission checking, once a practical assessment or assignment is submitted the lecturer needs tools that can check the integrity of the submissions. In this context the integrity of an assignment refers to whether or not the students did their own work, and did not copy, either from each other, or from other sources.

Checking for the “other sources” referred to in this article is difficult as this would entail the checking of all submissions against publicly-available resources. With written assignments this is easier as a search (using a search engine such as Google) of the web with selected text from an assignment could return a result [6]. With practical assignments it is much more difficult as the assignments can be stored in compressed folders on the web. Although checking practical assignments against publicly-available

resources is difficult in our opinion if a student gets a project from the Internet it is very likely that other students will also come across it on the Internet. Therefore when the submissions are checked this tends to come out. The only way of avoiding having too many other sources to check for is to make practicals unique to a course. Modern on-time, automated plagiarisms checkers are available but is mostly intended for written assignments (not programming assignments) and is not integrated into the submission facility of e-learning systems.

### 4. Conclusion

As was shown in this article there is a problem with online submissions. Post checking should be integrated into e-learning systems to minimise the corrupt use of e-learning systems. If it is not integrated into e-learning systems corruption is rampant, as was shown from the statistics presented in this paper. The e-learning system should also have integrated integrity checking that can be used by lecturers in an easily manageable interface. These tools are lacking in current commercial products and therefore needs to be integrated in such products.

### 5. References

- [1] WebCT Services, Authentication Integration, website, [http://www.webct.com/services/viewpage?name=services\\_authentication](http://www.webct.com/services/viewpage?name=services_authentication), Accessed on 8 August 2005.
- [2] Joy, M., Luck, M., Plagiarism in Programming Assignments, IEEE Transactions on Education, pp 129-133, 1999.
- [3] Culwin, F., Lancaster, T., Plagiarism Prevention, Deterrence & Detection, <http://www.ilt.sc.uk/resources/Culwin-Lancaster.htm>, 2001.
- [4] The Plagiarism Resource Site Charlottesville, University of Virginia, <http://plagiarism.phys.virginia.edu/links.html>, Accessed 19 October 2005.
- [5] Brin, S., Davis, J., Garcia-Molina, H., Copy detection mechanisms for digital documents, In Proceedings of the ACM SIGMOD Conference, pages 398-409, 1995.
- [6] McCullough, M., Holmberg, M., Using the Google search engine to detect word-for-word plagiarism in master's theses: a preliminary study, College Student Journal, [http://www.findarticles.com/p/articles/mi\\_m0FCR/is\\_3\\_39/ai\\_n15384389](http://www.findarticles.com/p/articles/mi_m0FCR/is_3_39/ai_n15384389), September 2005.