Source Code Plagiarism

Conference Paper · July 2009

DOI: 10.1109/ITI.2009.5196127 · Source: IEEE Xplore

Branko Kaucic
INITUT, Institut of information technology, Slovenia

Source Code Plagiarism

Dejan Sraka, Branko Kaučič
Faculty of Education, Kardeljeva pl. 16, 1000 Ljubljana
dejan.sraka@guest.arnes.si, branko.kaucic@pef.uni-lj.si

Abstract. *Learning programming languages and developing software is important part of curricula at many educational institutions. Directly connected to that is a peer-to-peer sharing of program's source codes which is not always allowed.*
The article discusses the problem of plagiarism, especially the source code plagiarism and presents an excerpt of the results of the survey among students at Faculty of Education and Faculty of Computer and Information Science at University of Ljubljana and among students at Faculty of Natural Sciences and Mathematics at University of Maribor. Results are revealing alarming situation. In addition, advices for the teachers to avoid the plagiarism as much as possible are presented.

Keywords. plagiarism, electronic plagiarism detection service, source code, programming, cut and paste culture

1. Introduction

Like the industrial revolution changed the way we live and work, the development of computers, the internet and web search engines changed the way of thinking. Massive amount of information is now accessible to everyone with access to the internet and relatively few skills are required for that. Traditional searching for information from the books could be seen as a thing from the past.

Because there is no way to forbid the users to use the internet, using it and online resources pose also some serious problems. Namely, the internet provides more conducive means to plagiarism which plagues to many disciplines. Educational sphere is not immune to that: usual education today is complemented with online resources, web classrooms, virtual worlds and easy online access to source documents. Many researchers report that plagiarism is increasing and presents a serious problem in education [6].

At present, at educational institutions there is a battle for a higher number of (good) students while the number of staff is not increasing. In addition, because of the incoming Bologna process, also the contact hours are reducing. The opportunities for a teacher to identify the students that need additional help will consequently decrease, and the only indicator if someone needs help will be the assessment results. Consequently, by increasing the number of students, the energy put into a single assessment will reduce (even more). Unfortunately, students are aware of that and some cheat (easier) because of that. Moreover, if assessment is copied from someone else and the teacher does not found that out, student ends the course with insufficient knowledge and can later in life have problems because of that. More worrisome, studies show that students not only plagiarize regularly but also believe that it is okay to do so [2].

Although, the temptation to plagiarize in the educational sphere is not limited to the students only, in this paper we restrict ourselves to the students only.

As in all similar faculties, because of the popularity of the ICT at Faculty of Education at University of Ljubljana we are also facing this problem. Consequently we started a project of studying plagiarism: why and how students plagiarize, studying of known plagiarism detection systems and developing a framework for professors and assistants that would detect plagiarism between different types of files. By understanding the background of reasons for plagiarism, we would like to develop electronic plagiarism detection services that will be useful in better way than generic systems. The second aim is to prepare advices and materials about plagiarism for teachers and students that would spread the following idea: *trust and student honesty must remain to obtain a successful academic system.*

The organization of the paper is the following. Section 2 presents the problem of plagiarism, how the plagiarism is noticeable in the programming courses, why some students do it, and some detection systems than can be used to detect it. In Section 3 we present an excerpt of a survey among students at Faculty of Education and Faculty of Computer and Information Science at University of Ljubljana and among students at Faculty of Natural Sciences and Mathematics at University of Maribor. Results are interesting and at the same time alarming. In Section 4 the paper is concluded and advices for reducing plagiarism are presented.

2. Plagiarism

The term "plagiarism" has many definitions. Encyclopedia Britannica [11] defines plagiarism as "*the act of the writings of another person and passing them odd as one's own. The fraudulence is closely related to forgery and piracy – practices generally in violation of copyright laws"*. Similarly, Webster's dictionary [15] defines plagiarism as *"a piece of writing that has been copied from someone else and is presented as being your own work."*

2.1. Plagiarism of programming source code

Researches show that plagiarism is on the rise [6]. It is a common problem in computer science courses. In many cases, the completion of programming assignments is a part of the course requirements.

Parker and Hamblen define source code plagiarism [10] as *"a program which has been produced from another program with a small number of routine transformations."* Source code plagiarism can vary from copy-pasting small amounts of program source code to copying large chunks of source code and masking everything with some techniques to disguise copied program. Possible modifications presented by Faidhi and Robinson [4] range in sophistication:

- level 1: changes in comments and indentation,
- level 2: changes of level 1 and changes in identifiers,
- level 3: changes of level 2 and changes in declarations,
- level 4: changes of level 3 and changes in program modules,
- level 5: changes of level 4 and changes in the program statements and
- level 6: changes of level 5 and changes in the decision logic.

For changes on level 1 little or almost none programming knowledge is required. On the other hand, since changes at level 6 require good programming knowledge and skills which actually prove that student master the programming, for the educational purposes it is more important to identify the changes on lower levels. Supported by the results of a survey in Section 3, additional level 0 at which no changes are made to copied source code, could also be added to the Faidhi's and Robinson's categorization.

With the Bologna reform, importance of grades for programming assignments on the student's final course grade is becoming even greater. Therefore, although sometimes team work is demanded, each student is usually expected to work independently on assignments. If instructors do not use any tools for automatic plagiarism detection or invest tremendous effort in reviewing students programming assessments, instances are found traditionally only on ad-hoc basis.

2.2. Reasons for plagiarism

There are many reasons why students use the work from each other, or collude when performing a specific piece of work. These include the following [14]:

- A weak student produces work in close collaboration with a colleague, in the belief that it is acceptable.
- A weak student copies, and then edits, a colleague's program, with or without the colleague's permission, hoping that this will go unnoticed.
- A poorly motivated (but not necessarily weak) student copies, and then edits, a colleague's program, with the intention of minimizing the work needed.

At the level of the individual student, three categories are often explaining why certain individuals commit non-trivial plagiarism [1]. The three categories concern students' personal circumstances, personal traits, and whether the means and opportunity to plagiarize are readily to hand:

Means and opportunity: the widespread of internet, and online academic journals have contributed much to the rising incidence of plagiarism, as they have made it possible for students to find and download materials from diverse sources with little reading, effort or originality. In addition, the web services with such materials are customized and thus difficult to detect using anti-plagiarism web crawling software. However, while the internet undoubtedly facilitates plagiarism, it does not possess the moral power to incite otherwise honest students to cheat. Lack of rules and prosecution for cases of plagiarism could encourage students to indulge in the practice.

Personal traits: Internal beliefs that academic cheating is immoral and dishonest are known to discourage plagiarism. However, the strongest motive for student cheating (according to Bjorklund and Wenestam's [1]) is the desire to obtain high grades, which itself may depend on other considerations. For example, due to a person's innate need to prove his or her worth to him or herself and/or to the world, or to a pathological fear of failure.

Individual circumstances: For example, students who need to take paid employment to help finance their time at university have less time for study, and high academic workloads may need to be compressed into their available study periods. The time pressures are likely to cause growing numbers of students to resort to plagiarism. It is also interesting, that males are more ready to admit plagiarism than females [1].

2.3. Plagiarism detection systems

For students who know about various instances of cheating, which instances are detected and which are not, the plagiarism is very tempting. The standard "dumb" attempt at cheating on a program assignment is to obtain a copy of a working program and then change statement spacing, variable names, prompts and comments. Comparing all pairs of solutions against each other for evidence of plagiarism seems like the approach that will detect this type of fraud.

However, even the above mentioned case is mostly enough to require a careful manual comparison, which simply becomes infeasible for large classes. Since there is usually more than just a few assignments, programming classes are in desperate need for an automated tool which performs reliable and objective detection of plagiarism.

In programming courses there are two sources of solved assignments: the internet and other students. For the internet, the products like Turnitin and PlagiServe are harnessing the internet for detecting plagiarism, in a similar fashion to what search engines such as Google has long been doing. However, the second source, other students, is more frequent and different plagiarism detection systems are needed.

Attempts to assess plagiarism, by technical means, run into the difficulty of distinguishing the differences between texts of different kinds. Much work in the past has been devoted to discovering concordances between texts. A plagiarism detection method must produce a measure that quantifies how close two source codes of programs are. Obviously, except for the case of a verbatim copy, detection approaches that use direct comparison of text files are weak, since there is no obvious closeness measure. Also, a simple file "diff" would of course detect only the most obvious attempts of fraud. There are various electronic systems to detect plagiarism in programming courses. From the middle of 70s to the end of 80s of 20th century the systems that were finding resemblances based on counting and comparing program attributes were prevailing. The technique was called attribute counting. Later, plagiarism detection systems that were examining and comparing program structures were introduced. Standard software metrics and examination of redundant code were used, too. There are even servers on the web which detect plagiarism. For example, JPlag [12] at Karlsruhe University tries to find pairs of similar programs, and the MOSS server at Berkeley [3] looks for similar code sequences in a set of programs; each system creates a web page where the instructor can see which ones are suspiciously similar. All of these techniques operate by running an analysis program on groups of submissions to detect similarities and to calculate the likelihood of plagiarism. Many approaches take a lexical approach, where the program tokens are classified as language keywords and user symbols. The simple plagiarism detection systems convert the source programs into token strings, and then compare the strings using dynamic programming. Although they are reasonably successful in pointing to pairs or groups who submit similar work, they are limited in identifying the original author of the work.

3. Survey on plagiarism

In this section we present an excerpt from a survey among students at Faculty of Education and Faculty of Computer and Information Science at University of Ljubljana and among students at Faculty of Natural Sciences and Mathematics at University of Maribor.

From 138 students that participated in the anonymous survey, 100 (72,5%) students admitted that they plagiarized at least once during their graduate study, either by writing programming assignment or writing a term paper, report, research or other assignment in the context of study obligations. Such high number of students was not expected and calls for measures.

Interesting is also the students' opinion about others. They estimate that 75,5% of all students plagiarized at least once. The highest estimation was 100% and the lowest was 5%. Standard deviation was 25% and median was 81,5% (Figure 1).

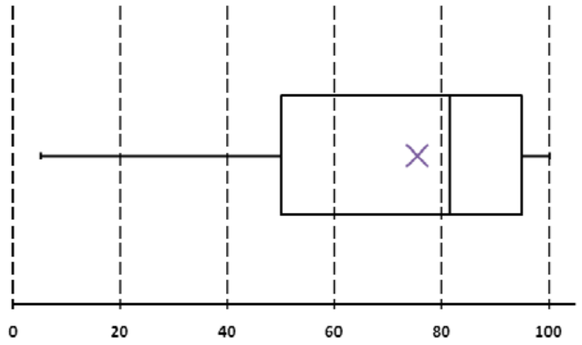

Figure 1: Box & whisker plot presenting students' estimation on spread of plagiarism

When students were asked about reasons for plagiarizing the source code, their responses were (it was multiple choice question):

Options	No. of answers	frequency
time pressure	51	61,4%
because everyone is plagiarizing	4	4,8%
to help a friend	25	30,1%
I could not withstand the amount of work	19	22,9%
because of external pressure to succeed	7	8,4%
I was not able to solve an assignment	58	69,9%
I was too lazy to solve it	13	15,7%
other	13	15,7%

Table 2: reasons for plagiarism

Students were also asked about their understanding of source code plagiarism by giving them different examples. 134 students thought that plagiarism in programming assignments is, when program includes:

situation	agree	disagree	undecided
program source of another program	90	11	33
comments in source code from another program	24	83	27
development plans from another program	71	25	38
documentation from another program	67	35	32
UI from another program	47	34	52
input data from another program	16	92	26

Table 2: understanding of plagiarism by examples

Only students from Faculty of Computer and Information Science and Faculty of Natural Sciences and Mathematics were asked about the changes they made in the source code when plagiarizing. From 56 respondents, 28 of them admitted, they have at least once plagiarized in programming course. When we questioned them about which changes they made in source code, the answers were (it was multiple choice question):

Options	No. of answers	frequency
no changes	3	3,6%
changes in comments and indentation	16	19,3%
changes in identifiers	21	25,3%
changes in declarations	14	16,9%

changes in program modules	10	12,0%
changes in program statements	6	7,2%
changes in program decision logic	6	7,2%
other	3	3,6%

Table 3: changes of source code made by students

At the answer "other" students listed different realization of subprograms, which could also be placed in category of changes made in program modules.

If we would like to reduce the plagiarism among students, we need to know how they can obtain solutions for their assignments. With internet and powerful search engines such as Google Code Search [5], Koders.com [9] and Krugle.org [7] they can search for solved assignments. Students can even hire an expert programmer for individual assignment through websites such as rentacoder.com. The most common sources of solved program assignments were students from current or previous generations. From 137 responders, 120 (87,6%) students admitted they have at least once given their solved programming assignment to other students. They usually provided complete programming assignments to other students through those media (it was multiple choice question):

Options	No. of answers	frequency
e-mail	106	88,3%
memory stick	47	39,2%
forum	8	6,7%
common repository	6	5,0%
orally with interpretation	70	58,3%
other	5	4,2%

Table 4: media used by plagiarists

At the answer "other" students listed Skype, MSN, CD and a paper with solved assignment.

We were also interested on the influence of professors and assistants on plagiarism. 78 students (53.4%) strongly agree with the statement that the professor and the assistant clearly stated that plagiarism is not desirable:

Likert scale	No. of answers	frequency
1 - strongly disagree	3	2%
2	16	11%
3	19	13%
4	30	21%
5 - strongly agree	78	53%

Table 5: role of professors and assistants

We also have to note that only 62 students (48%) of 129 respondents agreed with the statement that during their graduate study there was sufficient emphasis on learning proper quoting and referencing.

4. Conclusion

Plagiarism is a serious problem of today's copy-paste generation. It must be tackled at different levels: by using the plagiarism detection systems, proper regulations, educating students about plagiarism, and with proper assignments.

Teachers can use several systems for detecting possible plagiarism, as described in Section 2.3. However, they cannot be used efficiently if proper formal regulations do not exist. It is absolutely necessary to formally accept proper rules, regulations and procedures. At the same time they protect and guide the teachers when accusation is started, and the students before injustice accusation and sanctions.

In order to reduce plagiarism among students, we also have to educate. Teachers and students have to be educated about the importance of authorship, intellectual rights and rules of proper referencing and citing the resources. At programming courses, for the student it is better to cite from who the programming code is, without fear for the sanctions. In such case the teacher can identify the problems of understanding and if necessary help the student with additional explanation or by other methods. Proper method can be also pair programming which proved to be efficient and yields better results and consequently reduces the frequency of plagiarism [8].

For reducing the plagiarism teachers have to choose assignments that allow several interpretations and reduce the probability to obtain identical or semi-identical results. Each study year teachers should also change assignments and prevent reusing of source code between generations of students.

5. Literature

[1] Bennett R. Factors associated with student plagiarism in a post-1992 university. Assessment & Evaluation in Higher Education 2005, 30(2): 137-162

[2] Boden D, Holloway S. Learning about plagiarism through information literacy: A fusion of subject and information management teaching at Imperial College London. In: Plagiarism: Prevention, Practice and Policies 2004 Conference; 2004 Jun 28-30; St. James Park, Newcastle Upon Tyne. Newcastle: Northumbria University Press; 2004. p. 31-39.

[3] Bowyer W K, Hall O L. Experience Using ”MOSS” to Detect Cheating On Programming Assignments. In: Frontiers in Education Conference, FIE ’99, 29th Annual; 1999 Nov 10-13; San Juan, Puerto Rico. p. 13B3/18-13B3/22.

[4] Faidhi J A W, Robinson S K. An empirical approach for detecting similarity and plagiarism within a university programming environment. Computers and Education, 1987, 11(1): 11-19.

[5] Google Code Search. Google Inc.; 2009. http://www.google.com/codesearch [19.01.2009]

[6] Hammond M. Cyber plagiarism: are FE students getting away with words. Plagiarism: Prevention, Practice and Policies 2004 Conference; 2004 Jun 28-30; St. James Park, Newcastle Upon Tyne. Newcastle: Northumbria University Press; 2004. p. 257-264.

[7] Krugle Select. Krugle Inc.; 2009. http://krugle.org [19.01.2009]

[8] Nančovska Šerbec I, Kaučič B, Rugelj J. Pair programming as a modern method of teaching computer science. International journal: emerging technologies in learning 2008; 3(2): 45-49.

[9] Open Source Code Search Engine – Koders. Black Duck Software; 2009. http://www.koders.com [19.01.2009]

[10] Parker A, Hamblen J. Computer algorithms for plagiarism detection. IEEE Transactions on Education, 1989, 32(2): 94-99.

[11] plagiarism - Britannica Online Encyclopedia, http://www.britannica.com/EBchecked/ topic/462640/plagiarism [02/16/2009]

[12] Prechlet L, Malpohl G, Philippsen M. JPlag: Finding plagiarisms among a set of programs. Technical Report 2000-1. Karlsruhe: Fakultät für Informatik, Universität Karlsruhe; 2000.

[13] Rent A Coder: How Software Gets Done. Exhedra Solutions Inc.; 2009. http://www.rentacoder.com/RentACoder/ DotNet/default.aspx [19.01.2009]

[14] Schiller R M. E-Cheating: Electronic Plagiarism. Journal of the American Dietetic Association 2005; 105 (7): 1058-1062.

[15] Webster’s Online Dictionary, http:www.websters-online-dictionary.org/ definition/plagiarism [02/16/2009]