SOURCE-CODE PLAGIARISM: A UK ACADEMIC PERSPECTIVE

Georgina Cosma
University of Warwick
Department of Computer Science
Coventry, CV4 7AL
g.cosma@warwick.ac.uk

Mike Joy
University of Warwick
Department of Computer Science
Coventry, CV4 7AL
m.s.joy@warwick.ac.uk

ABSTRACT

We have conducted a survey of UK academics who teach programming on computing courses, in order to establish what is understood to constitute source-code plagiarism in an undergraduate context. The responses to the survey revealed that although there is a wide agreement between academics on the issue of what can constitute source-code plagiarism, some academics have expressed important issues concerned with source-code reuse, and self-plagiarism that need to be addressed in order to create a universally acceptable description on source-code plagiarism. In this paper we present the most important findings from the survey and suggest a description of what can constitute source-code plagiarism from a UK academic perspective.

Keywords
Source-code, plagiarism, reuse, self-plagiarism

1. INTRODUCTION

Plagiarism in programming assignments is an inevitable issue for most academics teaching programming. The Internet, the rising number of essay banks, and text-books are common sources used by students to obtain material, and these facilities make it easier for students to plagiarise. A recent article revealed that some students use the internet to hire expert coders to implement their programming assignments [6].

Bull *et al.* [1] and Culwin *et al*. [3] have carried out surveys on academics to determine the prevalence of plagiarism and have evaluated the performance of free-text plagiarism detection software and source-code plagiarism detection software respectively.

Subject Centre for Information and Computer Sciences

The surveys have shown that both free-text and source-code plagiarism are significant problems in academic institutions, and the study by Bull *et al.* [1] indicated that 50% of the 293 academics that participated in their survey felt that in recent years there has been an increase in plagiarism.

A review of the current literature on source-code plagiarism in student assignments reveals that there is no commonly agreed description of what constitutes source-code plagiarism from the perspective of academics who teach programming on computer courses. Some definitions on source-code plagiarism exist, but these appear to be very limited. For example, according to Faidhi and Robinson [5], plagiarism occurs when programming assignments are *"copied and transformed"* with very little effort from the students, whereas Joy and Luck [8] define plagiarism as *"unacknowledged copying of documents and programs."*

Furthermore, a book on academic misconduct written by Decoo [4], discusses various issues surrounding academic plagiarism. Decoo briefly discusses software plagiarism and the level of user-interface, content and source-code.

Sutherland-Smith [11] carried out a survey to gather the views of 11 teachers in the faculty of Business and Law at South-Coast University in Australia. The findings reveal varied perceptions on plagiarism between academics teaching the same subject, and the author suggests that a *"collaborative, cross-disciplinary re-thinking of plagiarism is needed."*

In order to establish what is understood to constitute source-code plagiarism in an undergraduate context we have carried out a survey that comprised questionnaires on this issue. In Sections 2 and 3 of the paper, we discuss the survey methodology and main findings respectively. Finally, in Section 4 we suggest a detailed description of what can constitute source-code plagiarism from a wide UK academic perspective.

2. METHODOLOGY

A web-based survey was created using the Questionmark Perception tool [11]. The web-link for the survey was distributed to a list of academics supplied by the Higher Education Academy subject centre for Information and Computing Sciences (HEA-ICS). The mailing list consisted of 120 names, many of whom can be assumed to have expertise in teaching programming. The people on the list were contacted in November 2005 by e-mail asking them to complete the questionnaire. Furthermore, the instructions for the survey specified that only academics who are currently teaching (or have previously taught) at least one programming subject should respond.

The survey was anonymous, but included a section in which the academics could optionally provide personal information. Of 59 responses, the 43 who provided the name of their academic institution were employed at 37 departments in 34 different institutions, of which 31 were English universities and three were Scottish universities.

The questionnaire contained mostly closed questions requiring multiple-choice responses. The questions were in the form of small scenarios describing various ways students have obtained, used, and acknowledged material. The respondents were required to select from a choice of responses the type of academic offence (if any) that in their opinion applied to each scenario. A comments box was placed below each question in order for academics to provide any comments they have about issues surrounding the question asked. It was very important to gather the comments of academics on the various issues regarding plagiarism due to the variety of academic regulations and the academics' opinions on such a sensitive issue.

It was not the purpose of this survey to address in depth subjective issues, such as plagiarism intent and plagiarism penalties, that could depend on student circumstances and university policies.

The data returned from the survey was very detailed, and this paper summarises the major results. A full report and statistical analysis is available elsewhere [2].

3. SURVEY RESULTS

In this section, we discuss issues raised by academics surrounding source-code plagiarism. These issues relate to source-code reuse and self-plagiarism that would otherwise not be an issue when considering assignments consisting of natural language text.

3.1 Source-Code Reuse

Academics were presented with four small scenarios on copying, adapting and converting source-code from one programming language to another, and using software for automatically generating source-code. The scenarios and the responses of academics for each of the scenarios are shown in Figure 1. Note, that academics were asked to provide their opinion as to which academic offence applies to each scenario by choosing their answer from the options provided. The options for each scenario were 'plagiarism', 'other academic offence', 'not an academic offence', and 'don't know'. Academics' comments raise important issues that are unique to source-code plagiarism.

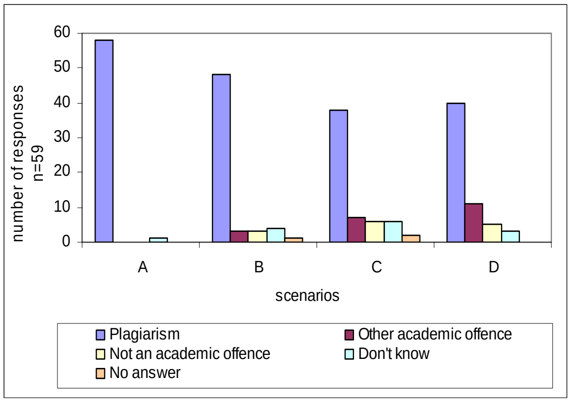

Figure 1: Responses to scenarios discussed in section 3.1

For the first scenario, A, 'a student reproduces/copies someone else's source-code without making any alterations and submits it without providing any acknowledgements' there was a wide agreement between academics (58 out of 59) that this scenario constitutes plagiarism. One academic has provided a 'don't know' response justified by the following comment.

"... in O-O environments where re-use is encouraged, obviously elements of re-use are not automatically plagiarism. I think I'd be clear on the boundaries and limits in any given circumstance, and would hope to be able to communicate that clarity to my students, but obviously there will potentially be problems. Use of the API would be legitimate without acknowledgement – or with only the implicit acknowledgement."

Regarding scenario B, 'A student reproduces/copies someone else's source-code, adapts the code to his/her own work and submits it without providing any acknowledgements' the majority of the academics have agreed that this action can constitute plagiarism. Many academics have commented that whether this scenario constitutes plagiarism depends on the *degree of adaptation* of the source-code, i.e., *how much* code is a copy of someone else's work and the extent to which that code has been adapted without acknowledgement. For example, beginning to write source-code using existing source-code and adapting it to such an extent that it is beyond all recognition, such that there is nothing left of the original code to acknowledge, then this may not constitute plagiarism.

However, two respondents were not clear on these points. The first, who did not provide an answer, noted that there were "*some grey areas here as software reuse is often encouraged.*" The second regarded the scenario as not being an academic offence, commenting:

"'This is difficult – as code copied from a website that assists in a specific task is potentially good practice. However, code that is a 100% copy is a different issue. I would also be concerned about the context of this copying. If the only deliverable where to be code and documentation the offence is clear. In this sense I suppose it is an issue of how much of the overall assignment is actually a copy of other work (without acknowledgement)."

For scenario C, 'A student converts an entire or part of someone else's source-code to a different programming language, and submits it without providing any acknowledgements', several academics remarked that if the code is converted automatically without any or much effort from the student then this can constitute plagiarism. However, if a student takes the ideas or inspirations from code written in another programming language, and creates the source-code entirely *"from scratch"*, then this is not likely to constitute plagiarism.

Furthermore, in their comments academics have pointed out that whether the conversion constitutes plagiarism depends on the programming languages, i.e., taking source-code written in one programming language and converting it to a similar programming language can constitute plagiarism, such as from C++ to Java, because the languages are too similar. However, converting Prolog to C or Java can still constitute plagiarism depending on the amount of work involved in the conversion. In addition, one academic who responded 'don't know', observed:

"*The key question is whether the student is being misleading about how much work is theirs or not. I can imagine examples where the translation was definitely plagiarism, and I can imagine examples where the student has taken legitimate inspiration from someone else's example code, and has rewritten it in a different language."*

A code-generator is an application that takes as input meta-data (i.e. a database schema) and creates source-code that is compliant with design patterns. An example of shareware code-generator software is JSPMaker [9], which given a database this software quickly and easily creates complete source-code and a full set of JavaServer Pages [7] that have database connectivity. We asked whether it constitutes plagiarism if 'a student uses code-generating software, removes the acknowledgement comments that were automatically placed into the code by the software, and submits it without providing any acknowledgements.' Academics have commented that this scenario can constitute plagiarism if the assignment specification instructs students to write the source-code themselves without the use of such software, or it would not constitute plagiarism if permission for use of code-generating software is described in an assignment specification. The majority of the academics considered *unacknowledged* use of such software as plagiarism. One academic who considered this scenario to be 'plagiarism' provided the following comment:

"In each case there must be some presumed benefit to the student in doing so (why did they do it otherwise?) and disruption to the assessment system. Even where the advantage might be minimal – e.g., from Prolog to C – a good student would almost certainly acknowledge the issue and use it to discuss the differences."

The findings suggest that whether or not source-code reuse is allowed in programming assignments, students should always indicate which parts of the source-code were not authored by them, and that using material created by other persons or by software *without providing acknowledgement* can constitute plagiarism, and as one academic commented, "*I require the students to acknowledge their dependence on these sources of code even when it is permitted."*

3.2 Self-Plagiarism in Source-Code

In student assignments, self-plagiarism occurs when a student copies entire or parts of his/her own assignment and submits it as part of another assignment without providing proper acknowledgement of this fact. However, when we asked academics whether it constitutes plagiarism if a student resubmits source-code they have originally created and submitted previously for another assignment we have received some controversial responses.

We gave the scenario 'assume that students were not allowed to resubmit material they had originally created and submitted previously for another assignment. For a graded assignment, a student has copied parts of source-code that s/he had produced for another assignment without acknowledging it' and asked the respondents – as before – to select the academic offence that, in their opinion, applies to this scenario. The responses are shown in Figure 2.

The majority of the academics (48 out of 59) have characterised this scenario as an academic offence (17 as plagiarism and 31 as other academic offence). In their comments, those academics have characterised this scenario as *"self-plagiarism"*, *"breach of assignment regulations if resubmission is not allowed",* and *"fraud if resubmission is not acknowledged".*

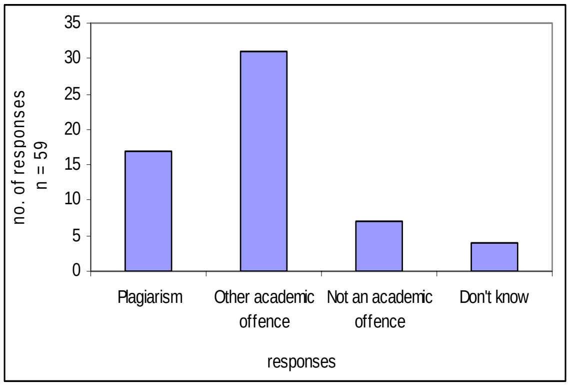

Figure 2: Responses to scenario on self-plagiarism

Some academics consider reusing source-code from other assignments and not providing acknowledgements as 'not an academic offence'. Those academics argue that in object-oriented environments where reuse is encouraged, it seems inappropriate to disallow students from reusing source-code they have produced as part of another programming assignment. The comments and responses provided by the academics who have not classified this scenario as 'plagiarism' or 'another academic offence' point to the controversial issue on source-code reuse aforementioned. One academic who replied 'don't know' remarked that *"Students should reuse code for assessments where possible!"* and another was clear that it was 'not an academic offence', and emphasised *"I find it hard to assume that students were not allowed to resubmit material."* A third academic, who also stated that it was 'not an academic offence', commented: *"would this ever happen in a programming oriented module when we behove students not to reinvent the wheel?"*

In conclusion, since 48 out of 59 academics have characterised the action of resubmitting source-code produced as part of another assessment as a type of academic offence (plagiarism or other) we can constitute that resubmitting source-code without providing appropriate acknowledgements *may* lead to an academic offence if this is not allowed for the particular assignment.

4. WHAT CONSTITUTES SOURCE-CODE PLAGIARISM?

The information that was collected from the survey responses was analysed and collated to create a description of what constitutes source-code plagiarism from a wide academic perspective.

4.1 Source-Code Plagiarism

Source-code plagiarism in programming assignments can occur when a student re-uses (4.1.1) source-code authored by someone else by obtaining (4.1.2) the source-code either with or without the permission of the original author and intentionally or unintentionally not properly acknowledging (4.1.3) the borrowed source-code and submits it as his/her own work.

If a student reuses (4.1.1) source-code that s/he produced as part of another assessment (in which s/he has gained academic credit) without properly acknowledging (4.1.3) this fact, it can constitute self-plagiarism or another academic offence (name of academic offence depends on university regulations).

If a student reuses (4.1.1) source-code authored by someone else (or produced by that student as part of another assessment) and *provides acknowledgements* then this may constitute a breach of assignment regulations, and not plagiarism (or self-plagiarism).

4.1.1 Re-use

Reuse includes the following:

a. Reproducing/copying without making any alterations.

b. Reproducing/copying and minimally or moderately adapting it. Minimal or moderate adaptation occurs when the work submitted by the student still contains some of the original source-code.

c. Converting the whole or part of someone else's source-code to a different programming language. Whether this constitutes plagiarism depends on the similarity between the languages and the effort required by the student to do the conversion. If the student takes ideas and inspirations from source-code written in another programming language and the source-code is entirely authored by the student it may not constitute plagiarism.

d. Automatically generating source-code using code-generating software can constitute plagiarism if the use of such software is not explicitly allowed in the assignment specification.

4.1.2 Obtaining

Obtaining the source-code either with or without the permission of the original author includes:

a. A student pays another person (or a student on the same module) to create part or whole of their source-code.
b. A student steals another student's source-code.
c. Two or more students collaborate (work together) on a programming assignment that

requires students to work individually and the students submit similar source-codes. This may constitute plagiarism or collusion (name of academic offence depends on the academic regulations).

d. Students between different groups carrying out the same assignment exchange parts of source-code with or without the consent of their fellow group members.

In the above list, source-code plagiarism can co-occur with other academic offences (such as theft, cheating, and collusion) depending on academic regulations. This is a very limited list since there are numerous ways that students can obtain source-code written by other authors.

4.1.3 Not properly acknowledging

Not properly acknowledging includes the following:

a. Not acknowledging the source and authorship of the source-code, within the program source-code (in the format of a comment) and in the appropriate documentation.

b. Providing *pretend* references (i.e. references that were made-up by the student and that do not exist) is a form of academic misconduct, often referred to as fabrication, and it can co-occur with plagiarism.

c. Providing *false* references (i.e. references exist but do not match the source-code that was copied) is a form of academic misconduct, often referred to as falsification, and it can co-occur with plagiarism.

d. Modifying the program output to make it seem as if the program works when it is not working, is a form of academic misconduct (i.e., falsification), and it can co-occur with plagiarism.

5. CONCLUSION

There exists survey-based research regarding the prevalence of source-code plagiarism in academia. However, we are not aware of surveys on the issue of what constitutes source-code plagiarism in UK universities. In this paper we report the findings gathered from a survey we conducted in order to bring together the various perceptions of academics on this issue, and to suggest a universally acceptable description on what can constitute source-code plagiarism from a wide academic perspective.

The data gathered from the survey has revealed two important issues that have received controversial responses. These issues regard source-code reuse, and self-plagiarism. Due to reuse being encouraged in object-oriented programming languages, some academics have expressed different opinions as to whether reuse and source-code resubmission without acknowledgement constitutes plagiarism, and self-plagiarism respectively. On both issues the majority of academics have shared the common opinion that when reuse is permitted students should provide appropriate acknowledgement to the parts of the source-code written by other authors (or that the students have submitted as part of another assessment) otherwise these actions can be considered as plagiarism (or self-plagiarism).

In the final part of the paper, we suggest a description of what can constitute source-code plagiarism, which academics may find informative when considering issues surrounding source-code reuse and plagiarism in student assignments.

6. REFERENCES

[1] Bull J., et al., *Technical review of plagiarism detection software report,* CAA, University of Luton, (2001*).*

[2] Cosma Georgina and Joy Mike, *Source-code plagiarism: A UK academic perspective,* Research Report No. 422, Department of Computer Science, University of Warwick, (2006). http://www.dcs.warwick.ac.uk/reports/422.html

[3] Culwin F. MacLeod A. & Lancaster T., *Source-code Plagiarism in UK HE Computing Schools, Issues, Attitudes and Tools*, Technical Report No. SBU-CISM-01-02, South Bank University (2001).

[4] Decoo W., Crisis on campus: Confronting academic misconduct. MIT Press Cambridge, England (2002).ISBN:0-262-04201-0

[5] Faidhi and Robinson, An empirical approach for detecting program similarity within a university programming environment, *Computers and Education*, vol. 11, no. 1, pp. 11-19, (1987).

[6] Gomes L., Some Students Use Net To Hire Experts to Do Their School Work. *The Wall Street Journal [Online],* (2006). http://online.wsj.com/public/us

[7] JavaServer Pages Technology, Sun Microsystems. http://java.sun.com/products/jsp/

[8] Joy M. & Luck M., Plagiarism in Programming Assignments, *IEEE Transactions on Education*, vol. 42, no. 2, pp. 129-133, (1999).

[9] JSPMaker v1.0.1. [Computer Software]. e.World Technology Ltd.

[10] Questionmark Perception [Computer Software]. Questionmark. www.qmark.com

[11] Sutherland-Smith W., Pandora's box: academic perceptions of student plagiarism in writing. *Journal of English for Academic Purposes*, vol.4, pp. 83-95, (2005).