Plaggie: GNU-licensed Source Code Plagiarism Detection Engine for Java Exercises

Aleksi Ahtiainen
Helsinki University of Technology
P.O.Box 5400
FI-02015 HUT Finland
aleksi.ahtiainen@iki.fi

Sami Surakka[1]
Helsinki University of Technology
P.O.Box 5400
FI-02015 HUT Finland
sami.surakka@hut.fi

Mikko Rahikainen
Helsinki University of Technology
P.O.Box 5400
FI-02015 HUT Finland
mikko.rahikainen@iki.fi

.

ABSTRACT
A source code plagiarism detection engine Plaggie is presented. It is a stand-alone Java application that can be used to check Java programming exercises. Plaggie's functionality is similar with previously published JPlag web service but unlike JPlag, Plaggie must be installed locally and its source code is open. Apparently, Plaggie is the only open-source plagiarism detection engine for Java exercises.

Keywords
Cheating, computer-assisted instruction, Java, open source, plagiarism, source code plagiarism detection engine

1. INTRODUCTION
A source code plagiarism detection engine Plaggie is presented. It is a stand-alone Java application that can be used to check Java programming exercises. Java is a common language for introductory programming courses. McCauley and Manaris [5] reported that 56% of accredited undergraduate computer science programs in the USA expected to use Java as their first language during the academic year 2002–2003.

Several source code plagiarism detection engines exist (Section 2). After the literature review in 2002, we selected JPlag [6] and MOSS [1] out of seventeen engines to be tested more thoroughly. We found JPlag as being the most promising for the needs of our laboratory. However, JPlag could not make a difference between (a) program code that was programmed by students and (b) program code that was distributed to them as a part of an exercise. This was a problem because one of our laboratory's programming courses used heavily exercises where some program code was distributed to students. Therefore, we decided to program a JPlag-like detection engine.

This shortcoming is solved in the current version of JPlag. Thus, the most important or the only contribution of Plaggie is that its source code is open (see Section 2).

JPlag or MOSS is probably the most suitable alternative for most computer science departments but Plaggie might be more suitable in the following situations: (a) A computer science department does not want to take a slightest risk that confidential information about cheating is sent outside the institution by mistake. For example, a teaching assistant might forget to remove student names or identification numbers from the submissions before sending them to a web service. (b) Someone wishes to develop Plaggie further or integrate it with other computer-assisted or computer-managed instruction software already in use. (c) Someone wishes to compare different source code plagiarism detection engines in detail using white-box testing instead of black-box testing.

2. RELATED WORK
For brevity, only one general publication about cheating and one review about source code plagiarism detection engines are presented. Wagner's web page [13] is a general discussion about cheating in computer science education including practical advice. For example, he discussed whether hardline or compassionate strategy is better.

Lancaster and Culwin [4] compared eleven source code detection engines and recommended web-based services JPlag and MOSS. We found these two engines as most interesting as well. They classified nine out of eleven engines as local (p. 104) and wrote: "Institutions wishing to use an entirely localized detection solution, for instance to avoid issues when submitting student work to an external source, should consider downloading and installing YAP3…" (p. 114–115). Unfortunately for us, YAP3 cannot compare Java programs (p. 105).

Some detection engines compared by Lancaster and Culwin [4] are listed in Table 1. Only the engines that can be used to check Java exercises are included. We searched at the web whether the source code of an engine was available and open. The results of this search are presented in the table.

Table 1. Some properties of detection engines for Java exercises

Engine	Source code available?	Source code open?
Big Brother	Unclear	Unclear
DetectaCopias	Yes	No
JPlag	No	No
MOSS	No	No
Saxon	No	No
SIM	Yes	No

Lancaster and Culwin [4, p. 104] wrote about Big Brother's availability: "special arrangement" which probably means that the source code is not open and the program might be available via email, for example. The source codes of DetectaCopias and SIM are available at the web. However, DetectaCopias did not use tokenization [4, p. 106], and in August 2006, DetectaCopias' instructions for use and comments in the source code were in Spanish, and the source code did not include copyright or license information [10]. In October 2006, SIM did not include copyright or license information [2].

In addition to engines compared by Lancaster and Culwin, we tried but did not succeed in finding open-source engines that

[1] Surakka is the corresponding author

ISBN: 978-1-59593-970-8

can be used to check Java exercises. The main results of this search are presented next.

SourceForge.net is the world's largest open source software development web site. We found three related projects from SourceForge.net: The BOSS Online Submission System, Sunlight, and Plagiarism Prevention Tools for Teachers. BOSS is a course management system originally developed for programming courses [12]. The project status is 4 – Beta and the source files are available for download [9]. The plagiarism detection engine Sherlock was integrated into BOSS in 2002 [12]. However, Sherlock is targeted at text analysis [11].

According to SourgeForge.net [8], "Sunlight will be an umbrella project for an exploration of source code similarity detection techniques, although the immediate and initial goal is to produce a C++ similarity measurement tool to allow for the detection cheating." The development status is 1 – Planning; that is, the project does not distribute any program files.

The project Plagiarism Prevention Tools for Teachers is apparently targeted at text analysis because they wrote: "…verifying the originality of the papers." The development status is 2 – Pre-Alpha and the project does not distribute any program files. [7]

3. PLAGGIE'S FUNCTIONALITY AND AVAILABILITY

Plaggie can check programs that are written in Java 1.5 also known as Java 5. Plaggie is GNU-licensed and can be downloaded from the web page of our institution [3]. Plaggie has been tested in Linux. Installing to Windows should be possible but this has not been tested. The file README_PLAGGIE includes the installation instructions, description of the algorithm, the list of features, and more.

4. DISCUSSION

When detection and handling cheating in programming exercises is considered as a whole, most work is done *after* a detection engine finds suspicious-looking student programs. A teaching assistant and/or a lecturer read the student programs, students are asked for explanations, and so on. Based on our experience and discussions with others, Plaggie is used 10–40% of time and the rest takes 60–90%. Therefore, it has a little practical meaning that web services such as JPlag and MOSS are somewhat slower than locally installed engines such as Plaggie.

It is possible that some developers of detection engines do not distribute their engines via web because this might ease cheating. However, we agree with Grune and Huntjens who wrote [2]: "We are not afraid that students would try to tune their work to the similarity tester. We reckon if they can do that they can also do the exercise." A bigger risk is that some company already selling essays and theses to students starts offering solutions to programming exercises as well. For example, ThesisExpress.com advertises: "Our system is powered by Plagiarism-Finder from Germany." It might be a hard-to-tackle cheating service if a company such as ThesisExpress.com used an auction service like Rent A Coder (www.rentacoder.com) as a subcontractor.

Integrating Plaggie into BOSS course management tool would be an interesting direction of further work. After all, BOSS was originally developed for programming courses. Anyway, we will probably make no major changes to Plaggie in the near future because it works well enough for the needs of our laboratory.

5. ACKNOWLEDGEMENTS

We thank Doctor S. Schaeffer for initiating the project and supervising the work in 2002.

6. REFERENCES

[1] Bowyer, K.W., and Hall, L.O. Experience using 'MOSS' to detect cheating on programming assignments. In: *29th ASEE/IEEE Frontiers of Education Conference*, San Juan, Puerto Rico, pp. 18-22. 1999.

[2] Grune, D. *The software and text similarity tester SIM*. Retrieved on October 9, 2006, from the Vrije Universiteit Amsterdam web site: http://www.cs.vu.nl/~dick/sim.html.

[3] Helsinki University of Technology. *Plaggie: Download*. Available at the Helsinki University of Technology web site: http://www.cs.hut.fi/Software/Plaggie/. 2006.

[4] Lancaster, T., and Culwin, F. A Comparison of Source Code Plagiarism Detection Engines. *Computer Science Education, 14*, 2, pp. 101-117. 2004.

[5] McCauley, R., and Manaris, B. *Comprehensive Report on the 2001 Survey of Departments Offering CAC -accredited Degree Programs.* Technical report CoC/CS TR# 2002-9-1, Department of Computer Science, College of Charleston, 2002. Retrieved on February 11, 2004, from the College of Charleston web site: http://stono.cs.cofc.edu/~mccauley/survey/report2001/CompRep2001.pdf.

[6] Prechelt, L., Malpohl, M., and Phlippsen, M. JPlag: Finding plagiarism among a set of programs with JPlag. *Journal of Universal Computer Science, 8,* 11, pp. 1016-1038. 2002.

[7] SourgeForge.net. *Plagiarism Prevention Tools for Teachers.* Retrieved on October 14, 2006, from the SourgeForge.net web site: http://sourceforge.net/projects/turnitin.

[8] SourgeForge.net. *Sunlight: Source code similarity measure.* Retrieved on October 14, 2006, from the SourgeForge.net web site: http://sourceforge.net/projects/sunlight.

[9] SourgeForge.net. *The BOSS Online Submission System.* Retrieved on October 14, 2006, from the SourgeForge.net web site: http://sourceforge.net/projects/cobalt.

[10] University of Chile. *DetectaCopias 1.0*. Retrieved on August 25, 2006, from the University of Chile web site: http://www.dcc.uchile.cl/~rmeza/proyectos/detectaCopias/index.html. [In Spanish.]

[11] University of Sydney. *The Sherlock Plagiarism Detector*. Retrieved on October 14, 2006, from the University of Sydney web site: http://www.cs.usyd.edu.au/~scilect/sherlock/.

[12] University of Warwick. *History of BOSS*. Retrieved on October 14, 2006, from the University of Warwick web site: http://www.dcs.warwick.ac.uk/boss/history.html.

[13] Wagner, N.R. *Plagiarism by Student Programmers*. Retrieved on August 25, 2006, from the University of Texas at San Antonio web site: http://www.cs.utsa.edu/~wagner/pubs/plagiarism0.html. 2000.