The Necessity and Results of Autonomous Integrity Evaluation of CAD Files

Dr. Jeff Morris
Department of Core Engineering
Rensselaer Polytechnic Institute

Abstract

This paper describes the history, necessity, methods, and results in performing large-scale collection and comparisons of CAD files for originality over the past 12 years (24 semesters). Higher-educational STEM-focused institutions are finding it necessary to evaluate modeling skills with CAD software in a quicker and more consistent manner. However, increasing mobile computing power and higher data bandwidth foster an alarming ease that students may transgress the institutional, course, and/or ethical standards by duplicating assignments and submitting work that was wholly or partially created/submitted by another student. During a first-year 14-week CAD course, hundreds of students create and submit thousands of CAD files for evaluation. Prior to autonomous technology, manual evaluation of student assignments for plagiarism yielded an average indictment rate of 0.9% per semester, over 5 semesters. Automatic checking has increased this to 7.4%. A program has been written that interfaces with a CAD software to parse through tens of thousands of CAD file assignments in matter of minutes. The program extracts relevant file properties to a spreadsheet, compares the set of files against each other for originality, and flags any file and student names that have identical properties. Over 15 semesters, this method has yielded a 100% conviction rate in 261 cases from a total pool of 3,861 students. A procedure to present the indicted parties evidence, render judgment and sentencing in a condensed period will also be discussed. As engineering instructors, it is a necessary duty to ensure that students adhere to rigorous academic standards, and if not, to call attention to their folly. This method and program strives to that end.

Introduction

The CAD instructional market is finding it necessary to evaluate introductory two-dimensional and three-dimensional parametric modeling skills with its software in a quicker and more consistent manner. Within this Institute's first year CAD course, hundreds of students are enrolled, submitting thousands of files, for which there is only one course coordinator to evaluate their originality. Assignment submissions can seem identical when only viewed from NX's GUI, thereby making it difficult to visually detect if a file has been duplicated.

Further, mobile computing platforms are dominating within U.S. higher learning, with price- points decreasing and CAD-sufficient hardware becoming ubiquitous across the top vendors. This lends to the ability for each student to own and use their CAD-ready hardware anywhere and at any time. Unfortunately, this mobility creates environments where students can engage in illegal file transfers or have their work stolen from an unattended laptop. In addition, the rise of cloud-based platforms (e.g. Course Hero™) that host student's collegiate work provides easy access for unauthorized duplication.

Background

Of the dozens of CAD software packages on the market today, some have employed add-ins or third-party applications in accomplishing automatic grading, quality, and/or integrity checks (PTC, 2018). Garland Industries combs through user IDs and timestamps for similarities of SolidWorks parts in their API program (Garland, n.d.). Some instructors have created an API to run similar checks for SolidWorks (Johnson, 2018), (Guerci, 2003) and NX (Kirstukas, 2018) for introductory CAD courses. While Guerci's methods were never published, Kirstukas have claimed an evaluation speed of 3 seconds/file, and he and Johnson concluded their method has less than perfect detection rates. This work describes a quicker and more robust method to interrogate NX files that cannot be easily tampered by users.

Current Course Format

Introductory Graphics and CAD is a one-credit introductory course that meets once per week, 14 weeks, for 110 minutes per meeting. All students are required to have personally owned laptops with working CAD software (Siemens NX). The typical semester enrollment totals between 280 and 350 students that must be divided into 8 or 9 sections due to seating constraints of laptop-ready classrooms.

Most assignments are presented as standard drawings similar to Figure 1, with the shape, parameters, and orientation given to the student, and the student is asked to create and submit an NX file in a *portfolio* folder.

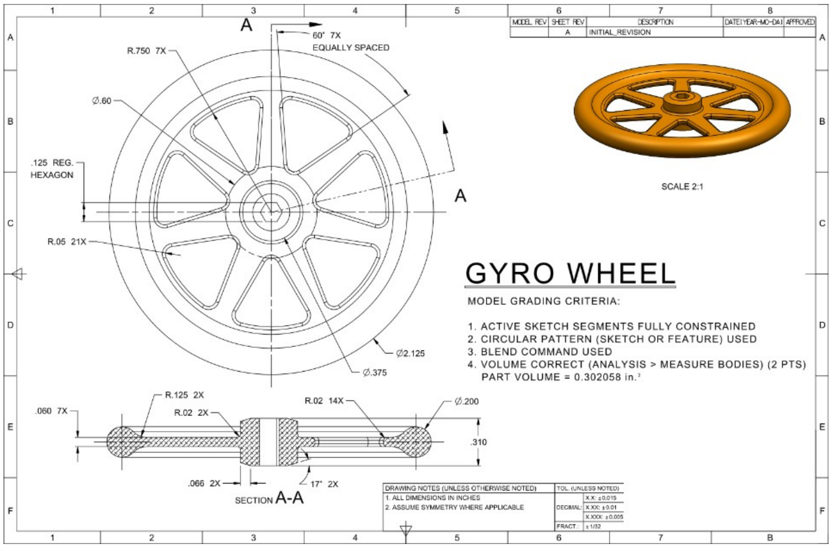


Figure 1: Example Model and Drafting Assignment

A student *portfolio* is defined as "the digital collection of every student-created file submitted for grade AND files obtained through the course learning management system necessary to complete all graded assignments". A complete portfolio is worth 10% of the course grade, and is submitted on the last day of the course.

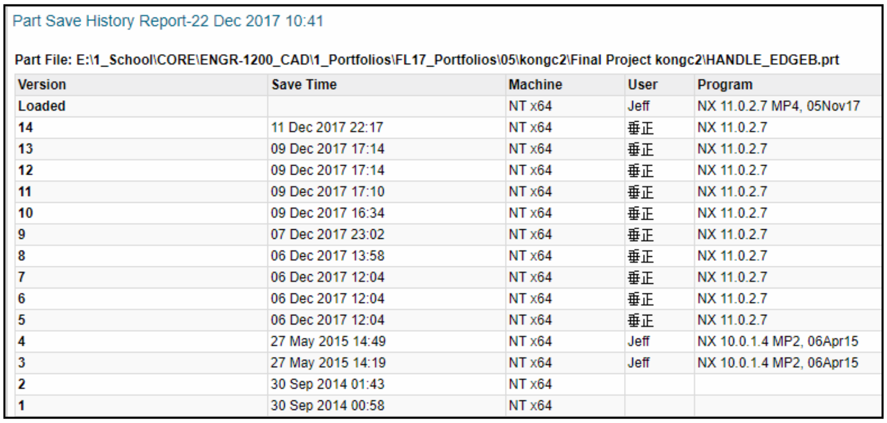
Complete portfolios will contain between 49 – 56 files, varying slightly each semester, totaling between 18 – 20 MB in disk space. The Spring 2018 semester contained 13,832 files within 252 portfolios submitted, totaling a disk space of 4.75 GB.

The syllabus contains a strict "zero-duplication" policy for any CAD model created in the course. Students are made aware that no work should ever leave their possession. Failing to adhere to the this will result in a failing course grade for all guilty parties involved and further disciplinary action if needed.

Integrity Evaluation Properties

User ID

The User ID listing in the file's Part History can list out the User logged into Windows at the time of save. Although using this parameter has yielded success by Kirstukas (2018), when student's use their own hardware, they will frequently have usernames (see the column **User** in below figure) that are not specific or meaningful for comparisons to other students, as denoted by an example student in Figure 2.



Part Save History Report-22 Dec 2017 10:41

Part File: E:\1_School\CORE\ENGR-1200_CAD\1_Portfolios\FL17_Portfolios\05\kongc2\Final Project kongc2\HANDLE_EDGEB.prt

Version	Save Time	Machine	User	Program
Loaded		NT x64	Jeff	NX 11.0.2.7 MP4, 05Nov17
14	11 Dec 2017 22:17	NT x64	垂正	NX 11.0.2.7
13	09 Dec 2017 17:14	NT x64	垂正	NX 11.0.2.7
12	09 Dec 2017 17:14	NT x64	垂正	NX 11.0.2.7
11	09 Dec 2017 17:10	NT x64	垂正	NX 11.0.2.7
10	09 Dec 2017 16:34	NT x64	垂正	NX 11.0.2.7
9	07 Dec 2017 23:02	NT x64	垂正	NX 11.0.2.7
8	06 Dec 2017 13:58	NT x64	垂正	NX 11.0.2.7
7	06 Dec 2017 12:04	NT x64	垂正	NX 11.0.2.7
6	06 Dec 2017 12:04	NT x64	垂正	NX 11.0.2.7
5	06 Dec 2017 12:04	NT x64	垂正	NX 11.0.2.7
4	27 May 2015 14:49	NT x64	Jeff	NX 10.0.1.4 MP2, 06Apr15
3	27 May 2015 14:19	NT x64	Jeff	NX 10.0.1.4 MP2, 06Apr15
2	30 Sep 2014 01:43	NT x64		
1	30 Sep 2014 00:58	NT x64		

Figure 2: NX Part History Information

Unique Part Identifier (UID)

From the Siemens NX Documentation: *starting with V10, each part is assigned a UID when it is created. The UID resides in the part file and is preserved for the life of the part – no matter how many times it is resaved or renamed in the operating system.*

The UID is a unique alphanumeric string that is generated for every part file created with the File > New (or Create > New in assemblies) command, even if custom template files are provided. It is this sole property that is checked for duplication across student submissions. No two student's submissions should ever contain the same UID. If so, it is plagiarism and must be flagged.

It is possible that a single student's portfolio contains several different CAD files with the same UID. This means the student duplicated a file and deleted and/or changed (i.e. 'rolled back') the features enough to build a different assignment. Both files are still the student's own original work and **not** indicative of plagiarism. While this method of creating 'new' files is strongly discouraged, it is never falsely flagged by this application. This application only checks for matching UIDs across different student IDs.

Timestamp

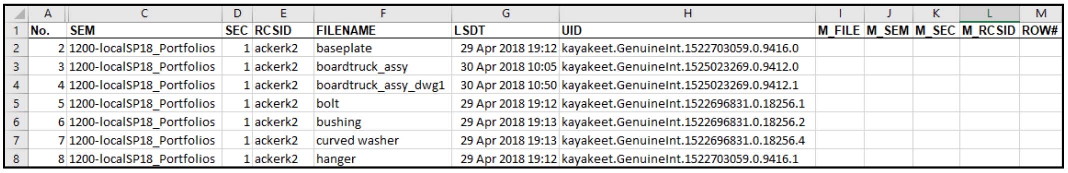
While timestamps are useful to extract self-plagiarism cases, this author currently allows users who have taken the course previously to re-submit older original files if they are the same assignment. The course has an extensive library of assignments that rotate every seven to eight semesters so the probability of old submissions is very low. However, a slight modification to the program could easily check for credible timestamps.

Integrity Evaluation Procedure

The author has written and tested an external .NET application that performs two separate routines to compare the originality of a set of CAD files.

Build Database

Figure 3 below shows an example partial output from a single student, indicating the file properties collected as column headings: semester taken, section number, student folder (the RCSID is a string unique to every student), filename, last timestamp the file was saved (LSDT), and UID. Each row is a separate file.



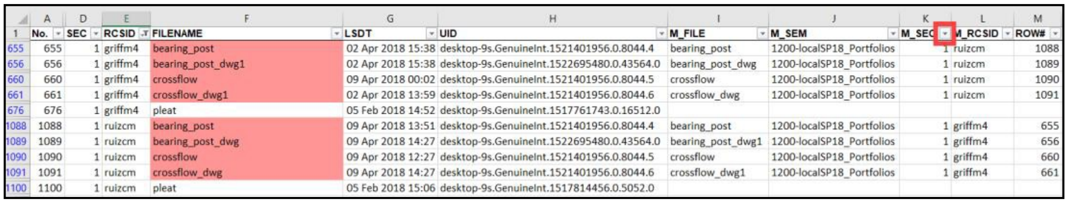
	A	C	D	E	F	G	H	I	J	K	L	M
1	No.	SEM	SEC	RCSID	FILENAME	LSDT	UID	M_FILE	M_SEM	M_SEC	M_RCSID	ROW#
2	2	1200-localSP18_Portfolios	1	ackerk2	baseplate	29 Apr 2018 19:12	kayakeet.GenuineInt.1522703059.0.9416.0					
3	3	1200-localSP18_Portfolios	1	ackerk2	boardtruck_assy	30 Apr 2018 10:05	kayakeet.GenuineInt.1525023269.0.9412.0					
4	4	1200-localSP18_Portfolios	1	ackerk2	boardtruck_assy_dwg1	30 Apr 2018 10:50	kayakeet.GenuineInt.1525023269.0.9412.1					
5	5	1200-localSP18_Portfolios	1	ackerk2	bolt	29 Apr 2018 19:12	kayakeet.GenuineInt.1522696831.0.18256.1					
6	6	1200-localSP18_Portfolios	1	ackerk2	bushing	29 Apr 2018 19:13	kayakeet.GenuineInt.1522696831.0.18256.2					
7	7	1200-localSP18_Portfolios	1	ackerk2	curved washer	29 Apr 2018 19:13	kayakeet.GenuineInt.1522696831.0.18256.4					
8	8	1200-localSP18_Portfolios	1	ackerk2	hanger	29 Apr 2018 19:12	kayakeet.GenuineInt.1522703059.0.9416.1					

Figure 3: Build Integrity List Output

Check Database

Figure 4 below displays an excerpt of the output after running a Check Database routine. If any UIDs are matched in the database, Columns I through M are now populated and filenames (column

F) are highlighted red. Four of the five shown files have been flagged as duplicated and shared. The “pleat” file is unique to both students, and is not flagged (not highlighted, and columns I – M remain blank since no match was found). The AutoFilter feature in Excel™ (denoted with small square in Figure 4) is used to sort and filter various properties.



	A	D	E	F	G	H	I	J	K	L	M
1	No.	SEC	RCSID	FILENAME	LSDT	UID	M_FILE	M_SEM	M_SEC	M_RCSID	ROW#
655	655	1	griffm4	bearing_post	02 Apr 2018 15:38	desktop-9s.GenuineInt.1521401956.0.8044.4	bearing_post	1200-localSP18_Portfolios	1	ruizcm	1088
656	656	1	griffm4	bearing_post_dwg1	02 Apr 2018 15:38	desktop-9s.GenuineInt.1522695480.0.43564.0	bearing_post_dwg	1200-localSP18_Portfolios	1	ruizcm	1089
660	660	1	griffm4	crossflow	09 Apr 2018 00:02	desktop-9s.GenuineInt.1521401956.0.8044.5	crossflow	1200-localSP18_Portfolios	1	ruizcm	1090
661	661	1	griffm4	crossflow_dwg1	02 Apr 2018 13:59	desktop-9s.GenuineInt.1521401956.0.8044.6	crossflow_dwg	1200-localSP18_Portfolios	1	ruizcm	1091
676	676	1	griffm4	pleat	05 Feb 2018 14:52	desktop-9s.GenuineInt.1517761743.0.16512.0					
1088	1088	1	ruizcm	bearing_post	09 Apr 2018 13:51	desktop-9s.GenuineInt.1521401956.0.8044.4	bearing_post	1200-localSP18_Portfolios	1	griffm4	655
1089	1089	1	ruizcm	bearing_post_dwg	09 Apr 2018 14:27	desktop-9s.GenuineInt.1522695480.0.43564.0	bearing_post_dwg1	1200-localSP18_Portfolios	1	griffm4	656
1090	1090	1	ruizcm	crossflow	09 Apr 2018 12:27	desktop-9s.GenuineInt.1521401956.0.8044.5	crossflow	1200-localSP18_Portfolios	1	griffm4	660
1091	1091	1	ruizcm	crossflow_dwg	09 Apr 2018 14:27	desktop-9s.GenuineInt.1521401956.0.8044.6	crossflow_dwg1	1200-localSP18_Portfolios	1	griffm4	661
1100	1100	1	ruizcm	pleat	05 Feb 2018 15:06	desktop-9s.GenuineInt.1517814456.0.5052.0					

Figure 4: Check Integrity Output

Speed

To manually check thousands of files each semester for plagiarism would be impossible. The build list evaluation performance is listed below in Table 1. The number of "Files Written" is different from "Files Processed" due to some internal filtering in the code; files that are given to the students from course staff (e.g. given parts for an assembly assignment) are filtered (using their UIDs) from being written to the database. These files are not created by the students and hence are not required to be checked with this tool.

Table 1: Program Speed Performance Summary

Type	*Semester*	*Sect.*	*Students*	*Files Processed*	*Time (h:mm:ss)*	*Files Written*	*Avg. s/Student*	*File/s*
Build Database	SP18	9	252	13,832	1:32:33	10,402	22.04	2.5
Build Database	SP17	9	248	12,754	0:55:52	9,499	13.52	3.8
Build Database	FL17	9	315	16,617	0:35:20	12,672	6.73	7.8
Check Database	SP18	9	252	10,402	0:01:43	10,402	0.41	21.6
Check Database	SP17	9	248	9,499	0:01:32	9,499	0.37	24.2
Check Database	FL17	9	315	12,672	0:02:33	12,672	0.49	62.1

The Spring 2018 semester included a total of 13,832 files submitted by 252 students. Of these, 10,402 files were student created. The entire check was completed in 94 minutes. Since *Build Database* routine runs approximately **10x slower** than the *Check Database* routine (due to opening and closing each file within NX for property extraction), the application separates them into two independent routines. The instructor can quickly accommodate a late portfolio submission, shown in Figure 5, without having to re-build the entire database. The instructor can simply run the Build Database routine for the late submission (~ 50 files) and append (copy/paste) the results into the previously built "master" spreadsheet that may contain tens of thousands of rows. The quicker *Check Database* routine is then performed again on the "master" spreadsheet.

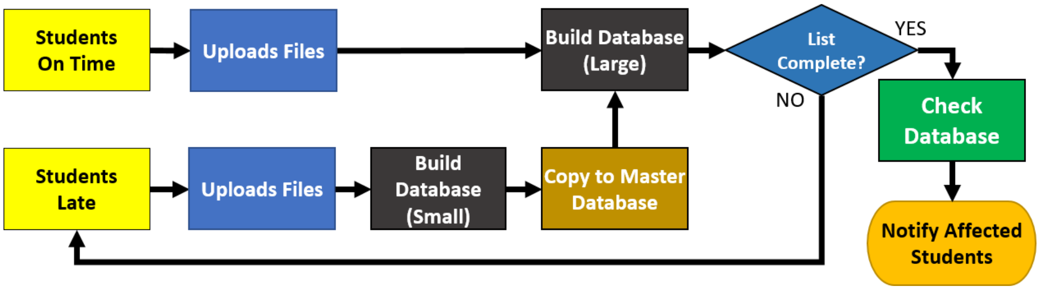


Figure 5: Current Integrity - Evaluation Procedure

Verification

Manual spot checking of plagiarism occurred until Spring 2009, when a Perl-based script was first used to extract compare User IDs. As mentioned earlier, using this method was less accurate, leading to missed false negatives or time-consuming manual investigation to filter out false positives. Table 2 summarizes the number of files flagged by the current .NET program versus the old Perl-based script.

Table 2: Program Comparison Over Three Semesters

Semester	Prog. Version	Sect.	Students	Files Written	Students Flagged	Files Flagged	False Negative	False Positive
SP18	NX 11.0.2.7	9	252	10,402	17	148	0	0
FL17	Perl	9	318	16,617	42	160	7	0
FL17	NX 12.0.2.3	9	318	12,672	42	167	0	0
SP17	Perl	9	248	12,754	14	144	0	64
SP17	NX 12.0.2.3	9	248	12,754	14	80	0	0

Validation

A conviction occurs when the indicted party enters a guilty plea or even in the cases of a not- guilty plea, fails to provide enough evidence that he/she is innocent. The vertical line indicates the implementation of automatic plagiarism detection between Fall 2008 and Spring 2009.

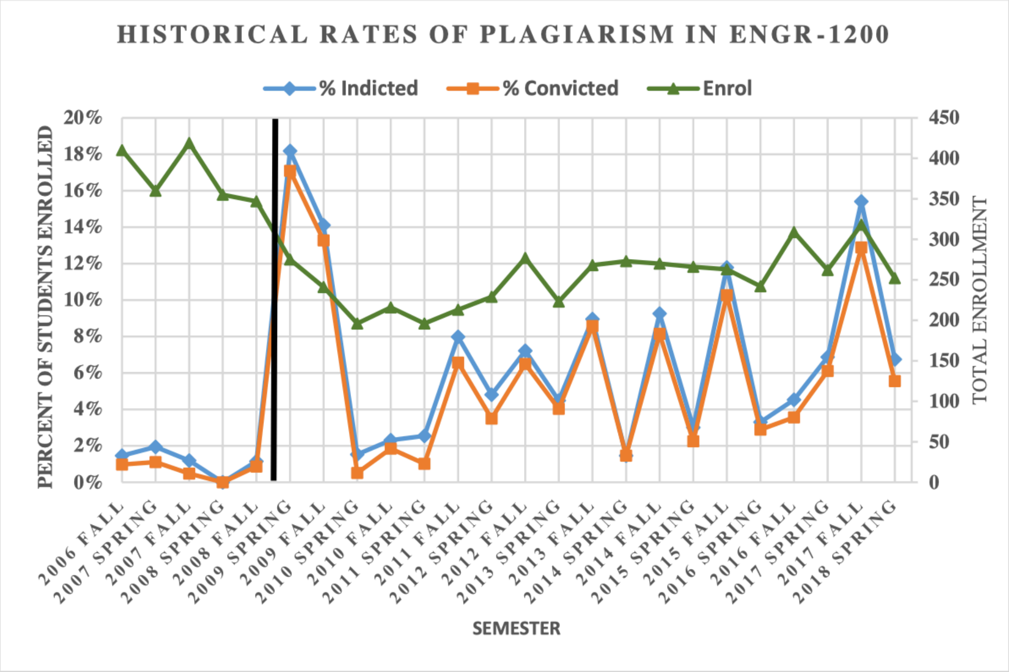


Figure 6: Plagiarism Detection History in ENGR-1200

A distinct decrease in the rate of indictment in the Spring semesters can be attributed to those students receiving warnings from peers enrolled in the Fall. Offenders usually enroll in the following semester and may share their warnings with others. Unfortunately, but unsurprisingly, the scenario seems to reset with each new crop of first-year students. From Figure 7, conviction rates have increased due to the following course changes: organized due process communicated on the syllabus, new assignments each semester, and an extermination of online download sites. In recent semesters, failed convictions are always due to an instance of theft (the victim’s charges are dropped), however the conviction rate among students who are “guilty” (by either of admission or lack of evidence to prove otherwise) has been 100%.

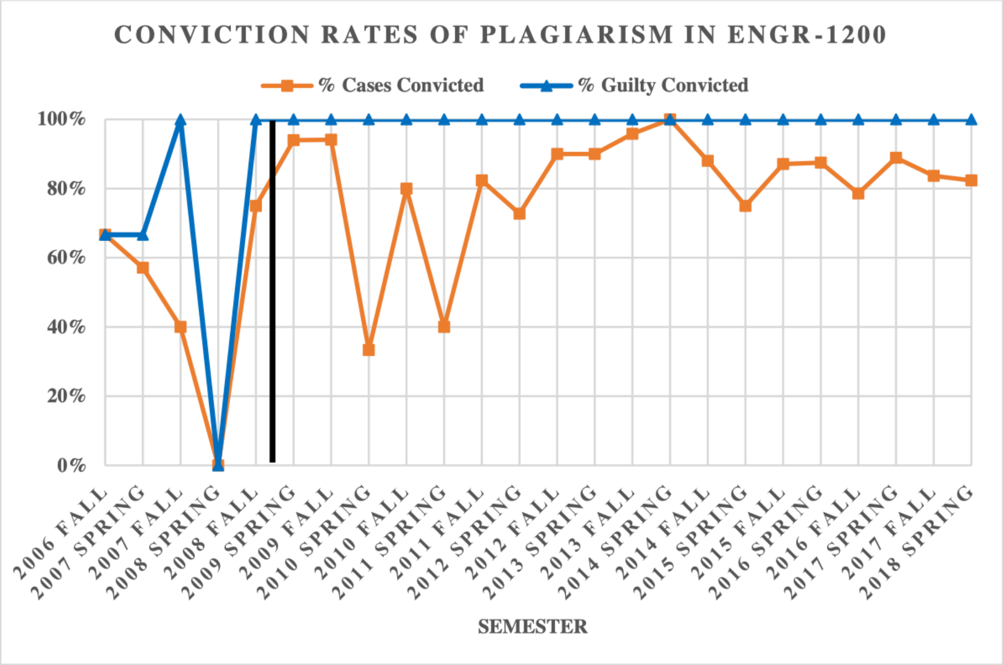


Figure 7: Conviction Rate History in ENGR-1200

Conclusions

One of the more difficult tasks in software education is ensuring the student creating and submitting the digital work are the same student. In cases of alleged theft, the coordinator must corroborate the testimony of both parties with the file "Part History" (to access which party originated the file). However, **there are currently NO applications, scripts, or programs that will guarantee that a student solely created the digital file they submit.** If a file never leaves a single machine, from creation until submission, there is no practical way to tell *who* in using the machine. However, this application demonstrates a 100% success rate of flagging instances of an NX file being duplicated, opened and saved on other machines, and submitted by two or more students. For those cases, automation is necessary for larger size classes submitting multiple files.

References

Garland, A. (n.d.). *GraderWorks*. (Garland Industries, LLC) Retrieved August 14, 2018, from https://garlandindustriesllc.com/index.php/pages/view/graderworks

Guerci, M. (2003). *Automated Grading System for Engineering Graphics and Computer Aided Design.* Troy: Rensselaer Polytechnic Institute.

Johnson, W. R. (2018). Detecting Plagiarism in SolidWorks CAD Courses. *ASEE Annual Conference & Exposition.* Salt Lake City: ASEE.

Kirstukas, S. J. (2016). Development and Evaluation of a Computer Program to Assess Student CAD Models. *Proceedings of the 2016 ASEE Annual Conference &*

Exposition. New Orleans, LA.

Kirstukas, S. J. (2018). Detection and Incidence of Plagiarism in a Solid Modeling Course. *Proceedings of the 2016 ASEE Annual Conference & Exposition.* Salt Lake City.

PTC. (2018, 24 2018). *Making the Most of CAD.* Retrieved August 24, 2018, from PTC University Expert Model Analysis: https://www.ptc.com/en/services/optimization/expert-model- analysis

Siemens Product Lifecycle Management Software Inc. (2018). Inspect Part Files - Report. Siemens PLM Software Inc.