A modular approach to automated grading of student assignments

Research into technology that can change the way assignments are graded.

Author:	*Marc M. Went*	**Date:**	*4 July 2014*
Study:	*Computer Science*	**First reader:**	*Ass. Prof. S. Voulgaris Ph.D.*
Level:	*Bachelor*	**University:**	*Vrije Universiteit, Amsterdam*

Contents

1. Abstract

Assignments are read, checked and graded manually, which is time consuming, costly and error prone.

This thesis investigates the feasibility of an automated grading system, selects the best approach and proofs its viability through an initial implementation.

The problem was studied in further details to better understand stakeholders and the multiple dimensions of the problem. Then research was done to investigate prior work in this area, and to recognize different solutions. The most viable solution was chosen, and implemented.

The conclusion is that a modular approach to automated grading of student assignments is both possible and economical, while at the same time adds lesser quantifiable benefits like teacher satisfaction and the possibility for student self-grading.

2. Introduction

All stakeholders of grading are negatively impacted by a manual grading approach. For universities it is a relatively high cost factor. Teachers spend a tedious amount of time reading, checking and grading assignments. As this is manual work, subjectivity creeps in and errors can and will be made. Students have to wait days and often weeks to receive their grade, and have the possibility to review them. By that time interest has faded to learn from mistakes or check the grading.

Automated grading addresses and resolves these problems by reducing cost and effort, replaces low rewarding work for teachers by more rewarding work, reduces grading time to minutes and allows self-grading by students when they rehearse on old assignments.

Different approaches exist to automate the process of grading assignments. Literature shows that typical approaches are point solutions, a program that grades a single assignment. Such an approach is prohibitive in time and cost to develop the grading solution.

A modular approach conceptually addresses these issues, but practically is much more difficult to implement. It has to take into account a great variety of assignments, from paper to electronic, from thesis to programming assignment, from style grading to solution grading.

The proposed solution in this thesis is a framework that can dynamically invoke modules for processing and grading assignments. For example, generic modules can be for OCR-processing – to convert written text to ASCII, for style checking – one module for each programming language. Assignment modules are specifically developed for a particular assignment. Grading is done by the framework by dynamically invoking the specific assignment module and possibly one or more generic modules.

Contributions of this thesis are:

A concise description of the problem (chapter 0,

- The problem / Claim)
- Research into earlier attempts to resolve the problem (chapter 4, Research)
- Investigation of different solutions (chapter 5, Possible solutions)
- Selection of the best possible solution (chapter 0,

- Chosen solution)
- Implementation of the solution to validate the approach (chapter 7, The implementation)

3. The problem / Claim

Introduction

Currently assignments are read, checked and graded manually. The consequence of this is that it is costly, time consuming, slow, not extensive, error prone and subjective.

No viable generic modular grading framework has been found that allows teachers to develop a specific grading module for the assignments they give to students.

Dimensions

Time & Money Grading is a necessity. When a teacher has a class of 50 students, grading easily takes 2 hours per student … This is also an expense to the university.

Duration Depending on the size of the assignment and the number of students, manual reading, checking and grading can take weeks. Students remain uncertain how well they did, and the longer it takes before they receive feedback the lower the impact on their learning performance.

Subjectivity & Errors As reading, checking and grading is done by people, it is impossible to prevent both subjective grading and mistakes being made. Either students accept this impact, or they don't, in this case a lot of time is wasted to debate these points.

Extensiveness In order to not spend too much time reading, checking and grading, teachers will do whatever is necessary, but nothing beyond that. It is in everybody's interest that assignments are as extensively checked as possible, with clear feedback what is "wrong" and why. This is delivered by an automatic grading system.

Self-grading Currently when students want to test their skills they can train on old assignments, yet there is no way for them to check how well they did. With an automated grading system, old assignments can be made and self-graded.

Stakeholders

Universities: Grading of assignments is a necessity and incurs substantial cost. This cost can be reduced through automation. When an automated grading system is built it will also attract more students, due to faster and more objective grading of their assignments and the possibility to train on older assignments and grade them.

Teachers: Grading of assignments is a relatively large portion of a teacher's time, which is often "outsourced" to teacher's assistants. Developing a grading module for an assignment requires less time and is more "fun" to do.

Students: Students have gotten used to the fact that grading takes time to complete, and hold a degree of subjectivity. A grade can be discussed with the teacher, but it takes considerable effort and with a small chance of success.

Claim

It is possible to develop a modular framework for automated grading of student assignments that addresses the problems of manual grading in a viable way.

4. Research

Introduction

The oldest reference to automated grading found is from 1994. Since then many attempts were made to conceive and develop such a system. This chapter researches previous work in this field, with the goal of building on such prior work.

Kassandra: The automatic grading system [1]

Built in 1994 this system is quite old. It uses a language called Maple to create a checking / testing module, knowing UNIX is a pre. This died out because knowledge of Maple is hard, especially since maple evolved to version 18 which is not necessarily backward compatible with the language it was 20 years ago. The UNIX command list is endless, to know sufficient about these commands a great effort must be put in, therefore it is not efficient to learn from scratch.

The principles that are elaborated in this paper can be reused for this project as well. The internal structure described in this paper is very useful since this is a well-organized way to implement a modular system.

Making automated building code checking a reality [2]

"*For automated code checking to be feasible, several standards and technologies need to be in place including the standardization of a building model …*" This paper argues that it is important to have a proper framework in place that allows these technologies to be set in place. Since this project is based around Computer Science, code checking will be a part that will be used a lot.

This paper is useful to see which standards are relevant towards code checking. These standard can be used when implementing the code checking module. Not only does this paper show where the relevance lies, but also how to prepare the project to take advantage of this code checking.

More Than Minutes: Teachers' Roles in Designing Homework [3]

This paper shows that when teachers create assignments with a specific objective, students will be more motivated to complete their assignments, which will result in better grades.

This research is a good example on how much time teachers spend on creating and grading assignments, as well as how much time students spend on assignments. This shows that an automated grading system is a necessary to invoke more studying with students. It also gives a proper indication on how much time is gained or lost by an automated grading system.

HWSAM: A Web-based automated homework submission system [4]

The system described in this paper is built in the early 1990's, it is the predecessor of many assignment submission systems. It is based as an alternative to emailing assignments, which even today is still common practice.

It is a foundational paper for the submission of assignments. Since it's been around for almost a quarter of a century, it can show the development of the internet. Yet is has not much relevance in current times, where the internet has evolved from use from time to time, to everyday use.

An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom [5]

With iGeom, geometry assignments are checked as soon as the students submits the assignment. This is a system that has been built to only solve those problems. Yet it shows it is possible to check geometry assignments using specialized algorithms.

This research shows that specific programs can grade difficult homework, yet it takes a lot of time to develop. This should be easier to be incorporated into modules, otherwise it is not cost-effective to have such an algorithm.

Automated Feedback for "Fill in the Gap" Programming Exercises [6]

This paper shows that with Fill in the Gap assignments code is executed on the student's machine whereas the code checking happens on the server. This code checking applies several filters to check how a program is set up.

This paper is a good example how to proceed on code checking, which can be used to determine how to define a proper module for assignment checking. Just as paper [2], this paper helps to prepare the project to make it easier to create code checking modules.

Homework: To Require or Not? Online Graded Homework and Student Achievement [7]

This study tests whether or not homework improves the grades of students. As well as automated versus manual checking of homework assignments. It shows that making homework assignments improve the score since the students study more.

This research shows where most of the time is spent, this can help the teachers design homework assignments. When done in conclusion with an automated checking the waiting time for the students can be shortened drastically. Therefore students can spend more time studying opposed to waiting for the results to come in.

Automatic grading of student's programming assignments: an interactive process and suite of programs [8]

The system described in this paper checks the programming assignment of students, it has checked over 20.000 programming assignments and it takes 4 to 8 hours to develop the assignment checking and the test plan. They compared this to manual checking of the assignments and achieved a 5% error rate on the automated checking.

This paper doesn't describe whether or not other assignments can be checked. Therefore it was of little use in this project.

Automatic Code Homework Grading Based on Concept Extraction [9]

This paper is very similar to the paper mentioned above [7]. They have made a program that the students must download to submit their homework to, this could be done much easier using the web.

Due to this paper, the conclusion can be made that some elements are not as profound as they could have been. Such as a dedicated program that must be downloaded before use. This helps this project to avoid problems such as these.

Automatic Essay Grading Using Text Categorization Techniques [10]

Assignments don't consist of only programming assignment, especially outside of computer science. Therefore it is important that modules should be able to check other kind of assignments.

This paper tests whether or not essays can be graded automatically, with great results. They score usually around the 80% correct. Not only does it show it is feasible, but also possible to implement with relative ease.

5. Possible solutions

Introduction

Based on prior research and relevant papers found, it becomes clear that there are several ways to tackle the problem. Holistically we recognize four options:

1. Manual grading of assignments
2. Point solution – without code re-use – for grading an assignment
3. Point solution – with code re-use – for grading an assignment
4. Modular solution to automated grading of assignments

Manual grading

The first possibility is not to do anything. The teacher will continue to check the assignments manually.

This has both advantages and disadvantages. The biggest advantage is that there is no need for change, students and teachers can continue the way they are used to. Yet this is also a disadvantage since the world around us is changing day by day, to continue an old habit in a new world is like bringing a sword to a gun fight.

Point solution – No code re-use

The first automatic grading option is to create custom grading programs per assignment. The teacher will have to create a program from scratch for every assignment.

The advantage of this approach is that it can be done quite extensive and precise. Yet it takes relatively a lot of time to develop. With many assignments it would take longer to make automatic grading systems than to check assignments manually.

Point solution – With code re-use

Re-using already existing code is very useful, one does not need to reinvent the wheel over and over again. Open source developers spend a lot of time to create code that does a certain task. This code is often tested quite extensively, and works well for the task it was designed for.

But some assignments require a square wheel rather than a round wheel. Changing this round wheel takes a lot of time and the results are uncertain. The same goes for re-using open source code as well. A programmer can change the code to suit his needs, but it is uncertain if this code will work as well as the original open source program did.

Modular solution

Last but not least would be creating a modular framework where grading modules would be created for every assignment given. The programmer does not need to create a framework for every assignment. Even older modules can be reused by newer modules. This increases the programming efficiency with every module created.

The time it takes to create a grading module is relatively low in comparison to manually checking assignments or creating custom grading programs. Yet creating a modular framework takes a lot of time. This time can only be justified when a lot of grading modules are created. The total development time will go down with every module that is created.

6. Chosen solution

Introduction

Developing a generic modular grading framework is a substantial investment in time and resources, which is a one-off investment, whereas the specific modules for grading become relatively easy to develop. A point solution is more work to develop than a grading module, but less than a generic modular grading framework.

If automatic grading is to be done just a few times, development of a generic modular grading system cannot be justified. However, given the number of universities and the number of students and the large number of assignments given to each student, the investment in a generic modular grading framework is easily justified (Recommendation #1).

For this project was chosen the modular solution.

Platform options

To create a proof of concept for a generic modular grading framework a programming platform has to be chosen.

One requirement is that the platform is web based. Students should be able to submit assignments from anywhere in the world without the need to download software to do so.

This eliminates programming languages like C, C++ and likewise. Even though these languages support the server model, it takes a long time to program a grading module.

Some popular server side programming languages are: PHP, JAVA, Python, Perl, Node.JS and ASP.NET. This thesis evaluates their suitability on the basis of several key features, which are explained below.

While Perl and ASP.NET were very popular at the beginning of the millennium, their popularity dropped drastically in favor of JAVA and Python. Therefore this thesis will focus on PHP, JAVA, Python and JavaScript.

JavaScript is best known as a client side hack and slash programming language where almost anything is permitted, and if you know the language well enough you can make some very cool stuff with it. But since 2009 JavaScript started on server side as well, this project is known as Node.JS. Therefore this platform is also included in the evaluation.

In the first quarter of 2014 Facebook released their own programming language known as Hack, this language is used as a substitute for PHP by Facebook. The switch between PHP and Hack is really easy since all PHP code will work in Hack since it is based on PHP. Given the influence of Facebook, Hack is also included in the evaluation.

Google on the other hand released in 2006 an extension to the JAVA programming language known as *Google Web Toolkit (GWT).* This extension allows web developers to create client and server side code in JAVA. The client side will be compiled into complex JavaScript and HTML pages that can be displayed to the user. Given the influence of Google, GWT is also included in the evaluation.

Platform choices

In "Appendix – Programming platforms" it is listed and argued which features are relevant for the choice of the best platform for developing a modular automated grading platform for student assignments.

All platform options have been rated against these features. The outcome of this analysis is shown in the table below.

Platform	PHP		Java		Python	JavaScript
Feature	*PHP 5.5*	*Hack Lang*	*Java Servlet*	*GWT*	*Python 2.7*	Node.JS
Latest release	May-14	May-14	May-13	May-14	May-14	May-14
IDE	Any	Terminal	Eclipse	Eclipse	Eclipse / Python IDE	Any
Debugger	XDebug	Terminal	Eclipse	Eclipse	Eclipse / Python IDE	Terminal
Type of Module	Class	Class	Package	Package	File	File
Learning curve	Easy	Normal if you know PHP	Normal	Hard	Easy	Hard
Speed	Normal	Good	Bad	Bad	Good	Good
Multithreading	Not native	Not Native	Native	Native	Native	None
Autoloading	Convenient	Convenient	Messy	Messy	Avoiding	Extra extension
Inline HTML layout	Easy	Hard	Not Available	Hard	Hard	Not Available
AJAX support	Good	Good	Difficult	Good	Good	Good
Code style	Clean	Clean	Clean	Chaos	Clean	Messy
	50	**39**	**28**	**22,5**	**37**	**29.5**

Table 1: Platform features versus Platforms

	Well implemented	100% score
	Acceptably implemented	50% score
	Poorly implemented	0% score

Conclusion

From the above table it can be seen that PHP, Hack Lang and Python are all viable choices for a platform, but **PHP 5.5 is the clear winner**.

It receives highest scores on all of the most important features, being "Latest release", "Learning curve", "Auto-loading"(Code snippet #1), and "Code style". The score is calculated by the weight each feature has. This table of calculation is described in further detail in "Appendix – Programming platform Selection criteria Weights".

7. The implementation

Introduction

The implementation of this project is the development of a modular framework that invokes two types of modules:

1. Generic modules (Code snippet #2)
2. Assignment grading modules

Generic modules are not specific to an assignment. A generic module to check code style of a program in a particular programming language is an example of such a module. See Appendix – Possible generic modules for a more extensive list of possible generic modules.

Assignment modules are developed by a teacher, to run inside the modular platform, to automatically grade a specific assignment.

Grading of an assignment is done through the modular platform by at least invoking the (specific) assignment grading module, and in addition possibly one or more generic modules.

The teacher will also have to login using their credentials. Modules, either generic or specific to an assignment, are all placed in the classes' directory. The modular framework will automatically detect the new module and load it into the system. (Code snippet #3)

The student will also have to login using their credentials. The modular framework will list all assignments that are due. The student drags and drops the assignment into the submission page, after which it will be automatically uploaded to the server, then checked by the appropriate module(s) and graded.

Initial start

Initially this project started with Hack language of Facebook as the programming language to tackle this problem, yet during the installation many problems arose. Hack was just barely released and was not widely supported. Also there was a big difference in operating system, Ubuntu 14.04 wasn't able to install Hack yet Ubuntu 13.10 was. When development started there was very little documentation on how to do specific non-PHP coding.

Back to basics

Because of all those setbacks Hack was switched for PHP using a plain Debian installation, without a hiccup. The first step was to set up an HTML page where students could drag and drop their assignments. Then it was important to send an AJAX request, instead of an old form request, so that the page does not refresh on receiving data.

Then the first step on the server side was to receive the file, save it and return a success. From there on the code would expand and very little would be done on the client side. Next the files needed to be accessed easily so a class that would handle all this was made. This would serve as a base for file handling and saving.

Modularity (Code snippet #4)

Next a temporary module was build that handles the test file. This means there is a base necessary that all modules follow, but within this framework they are free to do what they want. This framework performs generic functions – like saving the file to a specific location for the class, as well as checking if the set file extensions are acceptable, since one assignment requires another file than the other assignment. These are settings set within the module.

The module consists of two functions, a check function and a grading function. The checking function checks whether or not the file is empty or whether other not it contains a set minimum number of words. The grading function will proceed to grade the assignment. This is where everything boils down to. When code needs to be tested a Tester class is called that will try to execute the assignment, according to a defined test set made by the teacher.

The system could be extended to support multiple types of modules, this is beneficial for teachers who have none or almost no experience with programming (Recommendation #2).

It is up to the teacher how he will fill the module and whether or not he will let the module grade the student's assignment.

Backend

Modules are loaded using a native PHP function, autoloading, this makes it possible that even without knowing the exact location of the module, this module can still be loaded into the system. PHP supports since 5.3.0 Namespacing, this ensures that even when a module has a same name it can still be loaded (Recommendation #3).

On the backend of the module coding a lot of fail safes are built in so that even if the returned value is not what is to be expected, the code will still work. There can be for example a lot of specific results for tested assignments. If this predefined testing class would break there is no grading of the assignment.

The response each module has to make to the framework is always the same, this ensures that no matter what module is written, the response is always in the same way. This way the client side can read this response and handle it properly. Each response consists of a Grade and a Reason. The grade contains the grade the module calculated for the given assignment.

To calculate the result, each module can either create a custom comparison or use the default comparison. This custom comparison allows to compare two outputs in a different matter than the default does. This default compares whether the output is exactly similar, or contains certain keywords. A custom comparator has the ability to compare two outputs contextually, for example when the output is multithreaded and therefore the output inconsistent in order, but not in amount.

The reason can contain the explanation(s) why said grade was given. There can be multiple reasons why a grade was given. The code executed for example, but there was also found a high probability of plagiarism. Because of the second reason the low grade can be explained. It also gives the student support if they want to discuss the grade with the teacher.

To prevent students from submitting homework for one another, a login system must be in place. During the research the system called SurfConext came up, with this system students and teachers from all the universities from the Netherlands can log in (Recommendation #4).

8. Summary

Teachers spend a lot of time designing and grading assignments. For universities this implies cost and impacts the attractiveness of their university to both teachers and students. For teachers this implies a considerable time that cannot be spent on their core task, which is research. Also, grading is not the most exciting of work. As checking and grading is a manual task, results are subject and prone to error. For students it is tantalizing having to wait possibly weeks for their grade, and by that time receiving the results will not have that much educational impact anymore.

Previous research shows that automatic grading of assignments has been tried, yet many implementations are a single system. If for each assignment a single system has to be built to time to build this would rise above manual grading. Yet a system designed in 1994 has made steps to try to make a modular system. This system would use modules created per assignment to grade it.

The question arising is which solution to use to solve the problem of manual grading of assignments. First option would be to change nothing, yet the world around us changes very quickly and therefore change in the way assignments are graded is only natural. The second option would be to create a system for every single assignment. This would result in many different systems as well as a lot of development time for each individual system. Using already existing code could help this, yet only open source code could be used. But the most interesting option would be to create a modular framework where each assignment is a single module. These modules can be reused for newer assignments as well as used by students who want to test their skills on older assignments.

To create such a modular framework a programming platform must be chosen. Since everything happens on the internet it is only logical to create a webpage where the assignments can be submitted. Therefore only five platforms are interesting enough to consider. Considering all features relevant to such a platform, then PHP scores best. It is widely used and its development will not stop anytime soon. This ensures continuity for the modular framework being proposed and implemented by this bachelor project.

During the implementation there were several setbacks, first of all Hack did not work the way envisioned. Thus going back to start with PHP, with this system it was key to set up a proper modular framework. This started with the reading, writing and saving of submitted files. Next some generic modules were created to support this framework, these modules can be used by the assignment grading module as well.

Finally it was key to create a backend that could support any kind of module. This backend can be extended as well by using generic modules.

9. Conclusion

In the end the system works well for one module, yet more modules can be added. The system is set up in a way that would allow other open source code to be added as a module or as a generic module.

This project is far from done, the basis is set and can easily be expanded upon. This was a tremendous learning experience that helped to understand modularity much better. Not all systems are possible to be modular, or not workable because of the development time.

During this time, the internet age, it is surprising that no such system has yet been implemented – even though enough resources are available to create a proper working modular system, that is easy to implement for universities and colleges.

10. Recommendations

This bachelor project was limited in time (6 weeks) and resources (one student). Consequently areas remain to be investigated or developed further. This chapter recommends areas for such further work, possibly for other bachelor students to work on.

Recommendation #1. Investigate level of cost reduction through automated grading

These are estimates using previous experiences, this could be expanded upon by doing more research about the grading time and coding time of a module.

Recommendation #2. Investigate the possibility of generalizing grading modules

For example a generic module could be created that would allow different kind of grading modules. Currently grading modules are written in PHP but with such a generic module, an XML module could work just as well – this would make work for teachers without any programming knowledge to create modules as well.

Recommendation #3. Investigate whether or not it is possible to autoload namespaces

Autoloading of modules and classes could and should be improved upon. Currently the system assumes the classes that are the modules are unique. If there is a naming collision this would break the system. Therefore implementing namespaces into the autoloading is important, but since the current autoloading uses caches to store the classes it results in collisions. It should be investigates whether or not the namespaces can get the directories name, and if this can be added to the current autoloading system.

Recommendation #4. Investigate the security of the submission of homework assignments

A system called SurfConext allows students and teachers from all universities from the Netherlands to login and the system. There was insufficient time to implement this login system and test it properly. Yet this should be easy enough to implement. This would prevent students from submitting homework for one another, as well as teachers see for which classes they have to make modules.

Recommendation #5. Investigate the selection criteria and weights for a programming platform

Currently the criteria are weighted according to an opinion, it is good to research if this opinion is valid or that the weight need re-evaluation. If so why is this necessary and why it is better than the current estimates?

Recommendation #6. Investigate the expansion of generic modules

The current system has only a few generic modules, these need expanding to properly help development of more assignments.

Appendix – Programming platforms

Introduction

Many programming platforms exist in which an automated grading system can be developed. Developing a modular grading system puts higher demands on the choice of such a platform.

We have shortlisted six popular and well known options, which are described in some detail below. The choice for popular and well known platforms is fair, because a short learning curve for wide adoption and minimal effort are key requirements.

Java

The first language that is popular with developers around the world is JAVA, it is versatile in many ways and has been in development for almost two decades. The way in JAVA to make a web application is to use JAVA Servlets, these were popular during the HTML 1.0 period – they offered more content on a website, but was prone to leaks and was therefore replaced by Flash. It is a technology that still can be used but it very hard to master. You have to define every part of the window where the user would work in and the modules the teachers would have to make would have to be in JAVA. This is especially hard when the assignment is in a programming language any other than JAVA since it cannot run this code. There can be used difficult workarounds but they won't work as well.

GWT

A derivative of JAVA is Google Web Toolkit – or GWT for short – this is a JAVA library that is built to make development for Server and Client side web programming possible. With GWT you run a Tomcat server instead of any other server like you can with JAVA Servlets and you have to build a Client and Server side that are integrated with each other. Since this is a tight integration it is very hard to build modules for this JAVA library. As well as earlier mentioned argument that assignments of another programming language are hard to test since the main language is JAVA. Another problem that not only GWT but also JAVA have trouble with is Autoloading this isn't implemented natively and therefore when built it results in a really messy code.

JavaScript

JavaScript is well known for its client side programming capabilities, there are many libraries for the client side as well. But since 2009 JavaScript is also working on the server side, they have called it Node.JS and it is based on JavaScript. This is a very interesting possibility as well since many people, however closely related to web development, know at least something about JavaScript. Therefore the switch to Node.JS should be easy enough, yet the biggest drawback of JavaScript is that it doesn't support multithreading. This is a disadvantage when trying to test multiple assignments at once. The code will have to wait until the first assignment is done testing and grading. There is also no HTML inline possible with Node.JS since it runs only on the server and doesn't change any client side code.

Python

Python is a programming language that is often seen as the next generation for JAVA, yet is has been around longer than JAVA. Python has many built in libraries that help it to work as a webserver. It has a disadvantage that a module is written as a file, and does not require any structure like classes. It is not easy to create html pages with Python since it doesn't allow inline HTML layout. To automatically load modules autoloading is being avoided by Python. Therefore not properly usable.

PHP

Finally there is the most well know server side programming language for web development, PHP. This language has been around since 1995, like JAVA. It is a language that is heavily influenced by the C and C++ programming languages and by JAVA as well. Some people believe PHP is a language that belongs in the HTML 1.0/1.1 era, but it is still under heavy development and improvements and continually updating to work well in the HTML 2.0 era as well. Many developers only use the basics of PHP since it requires a weak and dynamic typing discipline, which can be hard for people who are used to JAVA or any similar languages. PHP has an extensive autoloading library which allows easier loading of modules. You can use PHP inside your HTML files as well which can help the developer create a seamless bridge between the client side and the server side.

Hack

Hack is a programming language developed by Facebook, it is based on PHP. The biggest difference is that the Typing discipline is neither static nor dynamic but it is both. This allows for code similar to JAVA but is executed on run time which allows for more dynamically created content. The biggest drawback currently is how to code Hack, since the only IDE that shows where the developer goes wrong is in the terminal. Any other IDE doesn't support code checking of Hack. Also HTML and Hack do not combine as well together as HTML and PHP. Only when XHP or a HTML template is used can Hack and HTML be combined.

Appendix – Programming platform Selection criteria

Introduction

Many programming platforms exist in which an automated grading system can be developed. Developing a modular grading system puts higher demands on the choice of such a platform.

We have shortlisted eleven criteria by which we judge each of the programming platform options. These criteria are described in some detail below.

Latest release

As Table 1 shows the latest releases are all in May 2014 except for Java Servlets, this is May 2013. Meaning that all those languages except Java Servlets are very recent and still in active development. When software is built, it is built to last. The odds of lasting are increased by using technologies that are under active development, since technologies that aren't updated anymore are prone to attacks on their vulnerabilities.

It is also important to assess whether a language will last or not, PHP, JAVA and Python all exist almost two decades and it is therefore save to assume they will last longer. Hack on the other hand is released in the first quarter of 2014 by Facebook, one can question how long Facebook will still exist and how long Hack will receive the funds to sustain active development. Node.JS is an odd ball in this group, it exists for five years now, but is under public development. When the public loses interest, no development will happen and the donations for development of this language will stop as well. But with the last two mentioned languages it is hard to assess if development will stop, or on the other hand they will prevail.

IDE

The type of IDE is also important to a language, when a language is bound to a single IDE it can limit the development speed of a programmer. Whereas languages that are not bound to a single IDE can be developed easier since the programmer can develop in their most preferred working environment.

Debugger

Debugging is essential to programming, all programs can be debugged using print statements to the console or terminal. This has a major disadvantage when the problem is buried deep in the code and one doesn't know where to look. Therefore a debugger is very important since it lets you test your code line by line. Usually IDE's include a debugger that let the programmer debug while coding. Sometimes a languages needs to be debugged using the terminal it runs in, this can be a little harder than any graphical debugging but is essentially the same.

Type of Module

For development of a module based program it is important to determine how a module is built. It is safe to assume there is a probability of naming collision between two modules of different semesters. Therefore a distinction between those files is very important. In Java a module is built as a package. This package has a specific name and a namespace, this namespace makes sure that the changes of naming collision are almost 0. Within PHP one has to write a class, this gives a higher probability of naming collision. Namespaces are available since PHP 5.3.0 but they don't work well with a very important feature, this feature will be discussed after this. Python and Node.JS both have modules that are based in one file. If a module has to be extended, this has to happen within that file to avoid even more collision based on file naming.

Learning curve

Another very important section for a languages is its learning curve. When a programmer is not used to a language it should be easy to get started in this language. Some languages are easier than others, especially when they are intuitive. Often languages like JAVA are widely known to application programmers and therefore easier to adapt, yet PHP is best known in the web programming and also most used. There are also plenty of examples online for widely known languages.

Speed

When designing a program that must handle a lot of request and page servings, it is important to look at the speed of the language. Not only the development speed but also the execution speed. Java is for example a fast language, yet experience learns that GWT is very slow. Both in development and in execution. On the other hand Python and Node.JS are quite fast and optimized for web development. PHP is neither slow nor fast, and if programmed correctly it can handle the requests just fine. Yet Hack had optimized its execution engine which gives it an edge over PHP in certain circumstances.

Multithreading

Speed can be improved by multithreading, systems can utilize multiple threads or cores of a processor to do calculations simultaneously. JAVA has a built in multithreading library that enables fast programming of safe multithreading code – multithreading adds a different dimension to programming since each thread should not override any other thread. Since Node.JS is essentially JavaScript, it doesn't support multithreading at all. To mimic these multithreading task, the programmer should write event driven code, which isn't always a possibility. Yet it isn't as important as a debugger is for example, one rather has a slow but functioning program than a fast nonfunctioning program.

Autoloading

A feature that makes modules much easier to implement is autoloading. This means the code doesn't need to know the explicit path to a module, but when a module is called the autoloading feature will search for this specific module. This feature is implemented very conveniently in PHP and Hack and can be extended if necessary with features like module caching. JAVA has the possibility to autoload packages but this requires a lot of coding and workarounds, it is by far as effective as PHP. Python tries to avoid the autoloading question at all cost. There are ways to implement this feature but they are very obscure and it is better to avoid this feature at all. Node.JS requires an extra extension since this feature is not native to the language.

Inline HTML layout

When designing a webpage it is easy to incorporate the language in the HTML page or rather the other way around. This is very easy within PHP, this language can mix and match with HTML within one file. Hack does not allow this kind of behavior, one has to write the HTML either in XHP or create an HTML template. This is not easy and takes a lot of time to learn properly. JAVA servlets don't allow this at all, essentially JAVA servlets are applications ran in a browser. GWT is similar but allows to create an HTML templates. Node.JS is only a server side language and therefore it must work with AJAX requests from the client.

AJAX support

As said above AJAX is very important, especially in Node.JS. Therefore the support of this is important, within PHP, Hack, GWT, Python and Node.JS AJAX is very well implemented and can be used with ease.

Code style

Last but not least, code styling, this is very important because when a programmer is pulled off a project and a new programmer steps in, then the code still must be easy to understand. GWT for example is utter chaos, even as a developer it is hard to keep track what part of the program is doing what. Whereas almost any other language has some kind of structure. Therefore it is important that a language organized and clean.

Appendix – Programming platform Selection criteria Weights

In the chapter on “

Chosen solution“ is described that PHP 5.5 is the best solution to implement modular platform for automated grading of assignments, based on relevant criteria.

Each criteria has been given a relative weight. These weights have been chose consciously, but are clearly debatable. However, that falls outside the scope of this thesis (Recommendation #5).

The table below justifies that conclusion, and has been included to illustrate the thinking process that led to the conclusion.

	Platform	PHP		Java		Python	JavaScript
Feature	**Importance**	*PHP 5.5*	*Hack Lang*	*Java Servlet*	*GWT*	*Python 2.7*	*Node JS*
Latest release	7	100%	100%	50%	100%	100%	100%
IDE	5	100%	0%	50%	50%	50%	50%
Debugger	4	0%	50%	100%	100%	100%	100%
Type of Module	3	50%	50%	100%	100%	50%	50%
Learning curve	10	100%	50%	50%	0%	50%	0%
Speed	4	50%	100%	0%	0%	100%	100%
Multithreading	3	50%	50%	100%	100%	100%	0%
Autoloading	8	100%	100%	0%	0%	0%	50%
Inline HTML layout	5	100%	0%	0%	0%	0%	0%
AJAX support	3	100%	100%	0%	100%	100%	100%
Code style	7	100%	100%	100%	0%	100%	50%
		50	**39**	**28**	**22.5**	**37**	**29.5**

Appendix – Possible generic modules

Recommendation #6

- **Word count**
  When a teachers gives an essay assignment where there is a minimum or maximum number of words, then this modules should help the Teacher check and enforce it.
- **Plagiarism**
  This is a very important yet difficult module, it must prevent students from copying each other’s homework.
- **Language detection**
  Check whether or not a preferred language is used.
- **Code function checking**
  This module could check whether or not enough functions are used, and whether or not they are used properly.
- **Style checking**
- **Programming language execution**
  There is currently a Java execution module (Code snippet #2) but there are many more programming languages that must be executed. These need a generic modules since there will be many assignments regarding programming languages and testing of these.

Appendix – Implementation logbook

20-March-2014
The first meeting for all the students who would do the Bachelor Project (Computer Science, IMM, and LI)

2-April-2014
The first meeting with Spyros Voulgaris, discussing the subject for this Bachelor Project. The first idea was to create a finger print scanner for elderly people's house, so they don't need a key. Spyros came with the idea of assignment checking program. This grew into a modular system that could check any kind of assignments.

3-April-2014
Read a lot of information about Hack Lang and HHVM, this is a language made by Facebook (*Appendix – Programming platforms: Hack*). Comparing Hack and PHP, and determining what the advantages are of Hack over PHP.

4-April-2014 – 9-April-2014
Trying to set up Hack and HHVM under several operating systems, Ubuntu 13.04, 13.10, 14.04b2, Debian 6 and 7. Yet none worked stable, it worked barely on Debian 7, but it couldn't serve ordinary HTML, CSS and JavaScript files.

10-April-2014
Ditched the idea of Hack and HHVM, it was a waste of resources. Then the client side programming started, first a basic drag and drop that would display the file in browser [11]. This was stripped of all the unnecessary code and images. Then server side started with receiving the document, and returning a success or fail.

11-April-2014
More debugging of the client server communication.

12-April-2014 – 19-April-2014
Started with a class that can handle files more appropriately than PHP, this class can give the path to the file, the name of the file, the extension of the file and many more function. Next was creating a base for modules, and at the same time a testing module in which I could test the base. This contained many trial and error, as well as thinking how to handle communication of any sort of module with the module base.

20-April-2014 – 28-April-2014
Since the beginning of the modules was created it was time to think how two modules would be handled. Therefore two classrooms were created, each had their own code – for testing purposed it was chosen to use: FFA-AAD and FFA-AAE (those are fictional codes). Each file uploaded would be put into the proper folder. And a message was sent if that class did not contain a preprogrammed module. Still improving the way grading and reasoning were handled. This was done with arrays but around the 28th of April two separate classes were created, one for the grade and one for the reasoning behind the grading.

29-April-2014 – 3-May-2014
Here the second autoloader was added for classes, the module autoloader already was working well, so a copy of the module autoloader was made for classes. Both autoloaders use a cache where the classes are saved to. This increases the speed modules can handle requests. During this period a lot of trial and error was done to get namespaces working in the autoloader, this feature was temporary abandoned because it took too much time. Also development of a tokenizer was started that could be used to see how a JAVA program was built – this could become part of a generic module, which could tell how many functions are used for example.

4-May-2014 – 10-May-2014
The finishing touches on the programming side are done, several exceptions are created so that the code can still run if something breaks. Also enumerators are added – these are non-native to PHP and therefore a minor workaround was necessary. This worked out very well and the code is a lot more rugged. The tokenizer was removed from the project since it took too much time to develop for a project with such a time frame.

11-May-2014
Finishing coding part, checking for major bugs and fixing them.

Appendix – Implementation code examples

Introduction

This autoloading function defines where this function should look for code, in this case in the modules folder. The save path for the cache file is also defined to improve the loading speed when more requests are done.

Code snippet #1. **Auto-loading** [12]

```
function module_autoloader($class) {

    $class = strtolower($class);

    $className = ltrim($class, '\\');
    $fileName  = '';
    if ($lastNsPos = strrpos($className, '\\')) {
        $namespace = substr($className, 0, $lastNsPos);
        $className = substr($className, $lastNsPos + 1);
        $fileName  = str_replace('\\', DIRECTORY_SEPARATOR, $namespace) .
DIRECTORY_SEPARATOR;
    }

    $class_filename = $fileName.'module.'.str_replace('_', DIRECTORY_SEPARATOR,
strtolower($className)).'.php';
    $document_root = $_SERVER['DOCUMENT_ROOT'];
    $class_root = "{$document_root}/php/modules/";
    $cache_file = "{$document_root}/php/cache/module.paths.cache";

    $path_cache = (file_exists($cache_file)) ?
unserialize(file_get_contents($cache_file)) : array();
    if (!is_array($path_cache))
        $path_cache = array();

    if (array_key_exists($class, $path_cache) && file_exists($path_cache[$class]))
        require_once $path_cache[$class];
    else {
        /* Determine the location of the file within the $class_root and, if found,
load and cache it */
        $directories = new RecursiveDirectoryIterator($class_root);
        foreach(new RecursiveIteratorIterator($directories) as $file) {
            if (strtolower($file->getFilename()) == $class_filename) {
                $full_path = $file->getRealPath();
                $path_cache[$class] = $full_path;
                require_once $full_path;
                $serialized_paths = serialize($path_cache);
                if ($serialized_paths != $path_cache)
                    file_put_contents($cache_file, $serialized_paths);
                break;
            }
        }
    }
}
```

Introduction

This code snippet shows how a generic modules looks like for programming assignments. All code must be compiled, this modules compiles using the exec function which allows to execute command line code. The execution function shows that whether or not the code could be executed. First of all the code must be able to compile to execute. The execution code contains 2>&1 which makes sure that even the error print statements are collected, normally the error statements go through a different channel.

Code snippet #2. Generic Java Module

```
public function compile() {
    if($this->compiled)
        return $this->compiled;
    $fileName = $this -> file -> getFileName();
    $compileCode = "javac $fileName.java";
    exec($this -> moveToDir . " && " . $compileCode, $output, $returnValue);
    $this -> compiled = $returnValue == 0;
    return $this -> compiled;
}

public function execute(array $args, $exact) {
    $this -> args = $args;
    $arguments = implode(" ", $this -> args);
    $fileName = $this -> file -> getFileName();
    $executeCode = "java $fileName $arguments 2>&1"; //echo stderr to stdout
    if ($this->compile()) {
        exec($this -> moveToDir . " && " . $executeCode, $output, $returnValue);
        if ($this -> executed = $returnValue == 0)
            $this -> output = $output;
        if($exact && Statics::arrayContains($this->output, "exception:", TRUE)){
            $this->executed = FALSE;
            throw new ExceptionException(implode(", ", $this->output));
        }
        return $this -> executed;
    }
    return FALSE;
}

public function closeClass(){
    $fileName = $this -> file -> getFileName();
    $removeCompile = "rm $fileName.class";
    exec($this -> moveToDir . " && " . $removeCompile);
}
```

Introduction

This is the main routine when an assignment is submitted, the class (As the 'ClassID') is send to the server as well. According to this class id the server knows where to look for a class that extends the ModuleBase. If there is no class found then an exception is thrown to indicate that the teacher has not created a module yet. If a class is found then the file is sent to this class for proper handling. When done with handling then a JSON parsed text is returned to the client.

Code snippet #3. Choosing the proper module for the submitted assignment

```
try{

    $class = $_POST['classID'];
    $folder = $_SERVER['DOCUMENT_ROOT']."php/modules/$class/";
    $className = Statics::getClassFromFolder($folder, "ModuleBase");
    if(!$className)
        throw new ClassNotInitializedException("There was no module created for
class: $class");

    $var = new $className($class, $_FILES['file'], $_POST['userID']);
    if ($var -> check())
        $json = json_encode($var -> grade());

    if($json)
        echo $json;

} catch(Exception $e){
    echo $e->getMessage();
}
```

Introduction

This is an example how an assignment module could be set up. The check function shows that the code checks the extension, whether or not the submitted file has the proper extension defined in the constant. It also checks whether or not the file has at least some content. Therefore the file size should be greater than 0.

Code snippet #4. An assignment module

```
class ModClass extends ModuleBase {

    protected static $ALLOWED_DOCUMENT_EXTENSION = ["java"];
    protected static $to_grade = true;

    public function grade() {

        $tester = new Tester($this->getFile(), "Java", realpath(__DIR__ .
"/test_case.xml"));
        $java = new Java($this -> getFile());

        $grade = new Grade();
        $reason = new Reason();
        $similarity = $java -> similarity(dirname($this -> getFile() ->
getPath())."/..") * 100; //Check if Java code is similar to any other in folder
        if ($tester->compiles()) {
            try{
                if($perc = $tester->testAll(TRUE)){
                    $grade->setGrade(8.0 * $perc);
                    $reason->addReason("Code output was " . round($perc * 100) . "%
similar to the expected output");
                } else {
                    $grade->setGrade(1.0);
                    $reason->addReason("Code couldn't execute given the
arguments");
                }
            } catch(TesterException $e){
                $grade->setGrade(1.0);
                $reason->addReason($e->getError());
                while(!is_null($e->getPrev())){
                    $e = $e->getPrevious();
                    $reason->addReason($e->getError());
                }

            }
        } else {
            $grade->setGrade(1.0);
            $reason->addReason("Code couldn't compile");
        }
        if ($similarity >= 90) {//almost same code
            $grade->setGrade(1.0);
            $reason->addReason("Code was " . round($similarity) . "% equal to other
code, very high suspicion of plagiarism");
        } else if ($similarity >= 75) {
            if($grade->getGrade() > 5.5)
                $grade->setGrade(5.5);
            $reason->addReason("Code was " . round($similarity) . "% equal to other
code, high chance of plagiarism");
        }
        return array("grade" => $grade->getGrade(), "reason" => $reason-
>getReasons());
    }

    public function check() {
        return $this -> checkExt() && $this -> getFileSize() > 0;
    }
}
```

Appendix – References

[1] U. v. Matt, "Kassandra: The automatic grading system," *SIGCUE Outlook,* vol. 22, no. 1, pp. 26-40, January 1994.

[2] C. S. Han, J. Kunz and K. H. Law, "Making Automated Building Code Checking A Reality," *Facility Management Journal,* pp. 22-28, September 1997.

[3] J. L. Epstein and F. L. V. Voorhis, "More Than Minutes: Teachers' Roles in Designing Homework," *Educational Psychologist 36,* pp. 181-193, 2001.

[4] S. Hsu, "HWSAM: A Web-based Automated Homework Submission System," *FIE'98,* November 1998.

[5] S. Isotani and L. d. O. Brandão, "An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom," *Computer & Education 51,* pp. 1283-1303, 2008.

[6] N. Truong, P. Roe and P. Bancroft, "Automated Feedback for "Fill in the Gap" Programming Exercises," in *Australasian Computing Education Conference*, Newcastle, Australia, 2005.

[7] Tisha L. N. Emerson and Kimberly D. Mencken, "Homework: To Require or Not? Online Graded Homework and Student Achievement ," *Perspectives on Economic Education,* vol. 7, no. 1, pp. 20-42, 2011.

[8] D. S. Morris, "Automatic grading of student's programming assignments: an interactive process and suite of programs," in *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference*, Boulder, Colorado, USA, 2003.

[9] I. Alhami and I. Alsmadi, "Automatic Code Homework Grading Based on Concept Extraction," *International Journal of Software Engineering and Its Applications,* vol. 5, no. 4, pp. 77-84, 2011.

[10] L. S. Larkey, "Automatic Essay Grading Using Text Categorization Techniques," in *SIGIR-98, 21st ACM International Conference on Research and Development*, Melbourne, Australia, 1998.

[11] R. Sharp, "Drag and drop, automatic upload," March 2012. [Online]. Available: http://html5demos.com/dnd-upload.

[12] M. Young, "Sparky San," 17 March 2011. [Online]. Available: http://www.sparky-san.com/efficient-convenient-autoloading-php/.

[13] R. d. Vrijer, "Handleiding Bachelorproject Informatica," November 2007. [Online]. Available: http://www.cs.vu.nl/~rdv/bp/bp-handleiding.html.