

# **Automatic Assessment in Undergraduate Level Engineering Drawing**

By

Rofans Belean Hilisebua Manao

Dissertation submitted in partial fulfillment of  
the requirements for the  
Bachelor of Technology (Hons)  
(Information & Communication Technology)

SEPTEMBER 2012

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar,  
31750 Tronoh  
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**Automatic Assessment in Undergraduate Level Engineering Drawing**

By

Rofans Beleam Hilisebua Manao

A project dissertation submitted to the  
Information Technology Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
BACHELOR OF TECHNOLOGY (Hons)  
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,

---

(MS. PENY GOH KIM NEE)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK  
September 2012

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements, and that the original work contained herein has not been undertaken or done by unspecified sources or persons.

---

(ROFANS BELEAM HILISEBUA MANAO

## **ABSTRACT**

AutoCAD is the most popular software used in undergraduate level Engineering Drawing education. Currently, there are no available methods to automate the marking process of AutoCAD assignments. This project is attempting to create a software prototype that can conceptually show how the marking process of AutoCAD assignments can be automated. The prototype will first convert the AutoCAD DXF file into SVG format and then evaluate each of the elements and attributes. The prototype is a web-based application and is able to achieve the aforementioned objectives. Given appropriate time and resources, this project can be extended into a full scale working application.

## **ACKNOWLEDGEMENT**

First and foremost, I would like to express my ultimate gratefulness onto Jesus Christ, my Lord and my Savior for his blessings upon me along the way of the creation of this project. I also would like to thank my parents who have been very supportive throughout my undergraduate study.

I also want to give special appreciation to my supervisors Ms Penny Goh and Ms Siti Rohkmah. Without their determination to guide me, I won't be able to finish this project.

At last, I would like to thank everyone who has helped me throughout this project. Without them, this project won't be able to finish.

## **LIST OF FIGURES**

|   |    |
|---|----|
| Figure 1: How the Software Shall Work   | 25 |
| Figure 2: Question 1 & Results          | 31 |
| Figure 3: Question 2 & Results          | 31 |
| Figure 4: Question 3 & Results          | 32 |
| Figure 5: Question 4 & Results          | 32 |
| Figure 6: Question 5 & Results          | 32 |
| Figure 7: Question 6 & Results          | 32 |
| Figure 8: Question 7 & Results          | 33 |
| Figure 9: Question 8 & Results          | 33 |
| Figure 10: Question 9 & Results         | 33 |
| Figure 11: Question 10 & Results        | 34 |
| Figure 12: Question 11 & Results        | 34 |
| Figure 13: Example of Circle Object     | 35 |
| Figure 14: DXF Representation of Circle | 35 |
| Figure 15: Conversion Methodology       | 37 |
| Figure 16: Homepage screenshot          | 38 |
| Figure 17: File Uploader screenshot     | 35 |
| Figure 18: Converter screenshot         | 37 |
| Figure 19: Marker screenshot            | 35 |

## **LIST OF TABLES**

|  |    |
|--|----|
| Table 1: Sample of DXF & SVG File                      | 27 |
| Table 2: Project Gantt Chart                           | 30 |
| Table 3: Comparison between student's and tutor answer | 38 |

## **ABBREVIATIONS AND NOMENCLATURES**

|     |                          |
|-----|--------------------------|
| SVG | Scalable Vector Graphics |
| DXF | Drawing Exchange Format  |
| CAD | Computer Aided Design    |

## TABLE OF CONTENT

|   |    |
|---|----|
| <b>Abstract</b>                               | 4  |
| <b>List of Figure</b>                         | 6  |
| <b>List of Table</b>                          | 6  |
| <b>Chapter 1: Introduction</b>                | 8  |
| 1.1 Background of Study                       | 8  |
| 1.2 Problem Statement                         | 9  |
| 1.3 Related Work                              | 10 |
| 1.4 Motivation of Study                       | 11 |
| 1.5 Aims & Objectives                         | 12 |
| 1.6 Scope of Study                            | 12 |
| <b>Chapter 2: Literature Review</b>           | 14 |
| 2.1 Introduction                              | 14 |
| 2.2 Automatic Online Assessment               | 15 |
| 2.3 Engineering Drawing                       | 19 |
| 2.4 DXF File Type                             | 20 |
| 2.5 Differential Conversion from DXF into SVG | 21 |
| <b>Chapter 3: Methodology</b>                 | 23 |
| 3.1 Introduction                              | 23 |
| 3.2 Research Activities                       | 24 |
| 3.3 Requirement Gathering                     | 25 |
| 3.4 Converter Module Development              | 26 |
| 3.5 Assessment Module Development             | 28 |
| 3.6 Delivery, Iteration, Post Development.    | 29 |
| 3.7 Gantt Chart                               | 29 |
| 3.8 Tools Required                            | 30 |
| 3.9 Key Milestone                             | 30 |
| <b>Chapter 4: Result and Discussion</b>       | 31 |
| 4.1 Survey Results                            | 31 |
| 4.2 Converter Module Prototype                | 34 |
| 4.3 Marker Module Development.                | 38 |
| 4.4 Prototype Screenshots                     | 39 |
| 4.5 Conclusion                                | 40 |
| <b>Chapter 5: Conclusion and Discussion</b>   | 41 |
| 5.1 Conclusion.                               | 41 |
| 5.2 Recommendation.                           | 43 |

## CHAPTER 1

### INTRODUCTION

#### 1.1 Background

Engineering Drawing is a type of drawing that in nature, used to fully and clearly define requirements for engineering items. It is usually created in accordance with standardized convention. Engineering Drawing is a platform for engineers to express their thoughts and ideas of concept involving geometrical shapes and design. It is important for virtual communication between draftsman and engineer, designer, architects and manufacturer, contractor, end-user and support service providers.

Starting from undergraduate level, engineering students are required to learn Engineering Drawing. But generally almost every engineering discipline requires good understanding on drawing engineering objects. There are variations over details on how to conduct the learning this course for different engineering discipline. The approach of learning this skill however can be achieved through variations of methods and tools.

In Universiti Teknologi PETRONAS for example, Engineering Drawing is a subject offered to students taking Chemical Engineering, Civil Engineering, & Mechanical Engineering. The main objective of the course is to teach the students the concept of Engineering Design by using Computer Aided Design tool. The main software used as Computer Aided Design tool in UTP Engineering Drawing course is AutoDesk AutoCAD. Students taking this course use that software to learn, exercise and create their assignments.



Participants of this course are required to submit their assignment files on a regular basis. Submissions are usually conducted by giving the files directly to the instructors on the lab session or mailing the files electronically. After a very inefficient process of collection, the course instructors are expected to manually assess each of the submitted files. With average participants of 300, a group of 4 lecturers and 8 tutors have to work very hard to mark a total of 7000 assignments every semester.

## 1.2 Problem statement

There hasn't been any software created to automatically assess Engineering Drawing course assignments at Universiti Teknologi PETRONAS. In fact there hasn't been any successful research done which **precise objective** is to automatically assess assignments of undergraduate level Engineering Drawing. In this situation, absence of such tool brings a lot of encumbrance to the course instructors.

Manual checking must be performed onto every single file of students' submissions. With a very large ratio between total numbers of instructors, this task seems to overburden the instructors. Therefore initiatives must be taken conduct a research that can produce a reliable software system that will resolve the aforementioned problems.

Furthermore we can elaborate the quest to be resolved by this project into a few points. First the software has to able to minimize human involvement in marking the Engineering Drawing assignments. In fact, if possible this software should be able to perform its tasks without any help from human operators. Second, the software should be able to reduce the amount of work and effort required to operate the software itself. This means that this software must be able to let its users to perform their task with speed. These tasks can be assignment submissions, marks organization, etc. Finally, due to the fact that the there are so many Engineering Drawing tools out there, it's important to design a software that is able to be extended or re-implemented with many different Engineering Drawing tools.

In this progress report, we have come up with answers of above problems. Due to the nature of its teaching, it's possible to come up with fully automatic assessment software for Engineering Drawing course. However, due the huge scope of the subject and limitation of time, we can't come up with fully working software capable to do the job for any Engineering Drawing files. However, we managed to come up with a set of algorithm and class design that shall be able to be used as a foundation of development of such software. The software is also developed as a web-based application to allow its users to operate it remotely. And in this research, we have decided to use AutoCAD as our companion tool, specifically its DXF file format. We also decided to utilize the open SVG graphic file format as another core element in this research. The rough idea on how these elements are used in this research is briefly presented in Section 1.5. Further elaborations and justifications for using these technologies are available in Section 1.6.

### **1.3 Related Work**

While there are no prior findings which objectives are exactly meant to create software to automate the assessment of AutoCAD assignments, there have been several independent yet interrelated works that will become a great input for this research. These related works are different in nature, yet somehow connected to this project. Those researches span from the online programming assignment assessor to the AutoCAD data conversion projects.

The natures of programming courses allow them to have online assessment scheme early on. There have been a lot of works done on this topic. Jackson (2000) proposed a combination of manual and automatic assessment to achieve such task. A fully automatic programming assignment system is presented by Saikkonen, Malmi, and Kohonen (2001). Despite fairly successful, both researches were still relying on the command line interfaces. A good example of GUI based online assessment system is BOSS that is created by Joy, Griffiths, & Boyath (2005). There is also a great discovery of software with similar objectives but is targeting different domain of undergraduate courses which was developed by Rasila, Harjula, & Zenger (2007). The software they created is meant for mathematics course. These independent yet interrelated works will give great inputs to the creation of the software from the

automatic assignment development point of view. This made possible through a great presentation of quantitative results on the effectiveness of their works and variety of methodologies that can be considered to be re-implemented in this research.

Other works that must also be taken into account is the works of reading AutoCAD data and files. A very detailed algorithm to read AutoCAD files is presented by Aslan, Seker, & Alpdemir (1999). There is also another decent work by Kachynkova (2005) that is talking about data transfer of CAD drawings. This is a fairly technical domain of work and requires great understanding of the nature of Engineering Drawing and its Computer Aided Design software counterpart. Incorporation of the findings of those researches was done selectively to ensure that they will still be relevant to this research's scope.

There are also researches done on SVG and how it will relate into AutoCAD DXF format. Ahmad & Haque (2001) presented a very good algorithm to read parts of DXF drawings and how to convert them into SVG. There is also a comparison between SVG and DXF performance by Fahiem & Farhan (2007).

#### **1.4 Motivation of Study**

The idea of automatic assignment software was first laid down upon the realization of the imbalanced students-instructors ratio. That situation created inefficiency and ineffectiveness in assessing the Engineering Drawing assignments. Therefore logic dictates that creation of software capable to get rid of that inefficiency and ineffectiveness shall be beneficial from a business point of view. This software once completed will be able to save the time, cost, and energy of the instructors and participants of Engineering Drawing course. Due to popularity of the course and the big numbers of student taking it, this research has a very good market potential.

Besides offering a promising business opportunity, this research is also able to stand out due to its uniqueness. Most people will misunderstand this project as an image processing based research. This is untrue due to the fact that Engineering Drawing files are not processed as image by their readers, although conversion from former to latter is possible. The research instead relies on the documentation created for SVG

& DXF format and strives to design a solution consistent to those files properties and nature. This off course can be achieved after fully understanding the way these two file formats can be related.

Since the SVG & DXF format is widely used and has a huge scope, it's hard to actually complete working software capable to cater to all of their attributes. The research therefore is trying to produce a working design that is modular and extendable that is working for a few of those formats properties. The software design is intended to also work with the other attributes of the file format once the design is extended. Here, a challenge remains as it's a well known fact that good software is hard to design.

In conclusion, the business potential, the novel approach, and the challenge of this research are the driving forces that motivate the birth of this project.

## **1.5 Aims and Objectives**

By the end of this project, we shall produce software (web based application) that is able to:

- Retrieve and read DXF files that contained simple objects (ex: Circle)
- Convert them into the SVG format (by ensuring no data loss in the process)
- Perform automatic assessment into the converted files by comparing each of the student's elements with tutor's elements
- Return the mark (score) of those files by calculating the stand deviation of the differences the elements

## **1.6 Scope of Study**

The task of doing automatic assessment in Engineering Drawing that is used in this research is basically breakable into three steps. First is reading the drawing inputs (DXF files), then converting them into other open and data interchangeable formats (SVG), and finally assessing the conversion products.

Reading the drawing inputs can be achieved by first creating a user interface to allow files upload activity. The uploaded files will later be read and understood by the software to determine its content. Later on, using an algorithm capable to transfer the drawing content into another format, the software will convert the files.

Afterwards, these files will be transferred from their original DXF format into SVG. Due to the variety of drawings, it's practically impossible to simply create a single algorithm to read all of them. The best approach is to create a variation of algorithm that is based on one standard algorithm that will work to suit different kind of drawings or shape in the drawings. The standard algorithm is basically reading the DXF files line by line and finding certain parts of the files in order to transfer them into the matching SVG format. The process to determine which parts to find and which SVG format it will be converted into will vary depending on the shapes and nature of the drawing.

The freshly produced SVG files will later be processed again. These files will be later matched to a file that is previously stored. This file is supposed to be the correct version of the assignment. The comparison is supposed to be done by comparing the assignment files and the correct file from elements to elements. However the comparison algorithm is yet to be produced. We have determined however that the algorithm will utilize XQuery or XPath language. These are the standard tools for reading XML based files (such as SVG).

All programming in this work is done in C# as elaborated in details at section 1.7.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

In this introduction part of the Literature Review, uniqueness of this project is going to be clarified. Specifically, this section will explain on why this project is not an Image Processing based research. Image Processing is not actually a wrong approach but it will consume more time, cost, & resources than the method proposed in this project.

There have been attempts to utilize image processing method to accomplish the goals set by this research. In her final year project, Othman (2007) a fellow UTP student suggested an Image Processing approach to automatically assess Engineering Drawing assignments. Image processing is generally defined as part of Signal Processing where Image acts as the input to the process. Othman assumes that Image Processing approach shall work as it is the commonly used method to deal with bitmap based graphics.

Through Image Processing Othman proposed to perform the object recognition which is somewhat related to artificial intelligence research domain. In depth, this task is achieved through converting the raster objects of the drawing files into perfect graphic objects by vectorization. After the conversion the image should be divided into several constituent regions or segments. Then afterwards each segment will be processed separately into some edges detection algorithm. Then the segments will be assessed through template matching sequence by comparing each segment with the correspondent segment of the correct answer.

However when in her research interim report, it was mentioned that Othman was not able to accomplish her research goals. The downfall of the research was mainly because of the imperfect approach it has used. Her realistic and valid method is not being supported by resources that she had. Therefore, she was unable to accomplish the project.

Now there are several important lessons that we can learn from Othman's work. First of all, she disregards the nature of DXF files which already supports edge detection and differential conversion. In fact, there are a great number of researches that have been done to perform that task. Instead, Othman proposed a brand new methodology which yet to be proved feasible. The research should instead focus on utilizing existing resources.

Another aspect that she seemed to leave out is the job of automatic assignment assessment is an educational issue. She focused her work into deep technical details without putting the equivalent effort to ensure that the automatic assessment system will be able to give the instructors and participants a good result. Well there have been no exact research completed on assessing engineering drawing on undergraduate level automatically; similar work has been implemented into programming and data structure courses. They should provide enough resource so that the research can be directed into achieving the technical goals along with the satisfying stakeholders' targets. Although her work seems to be incomplete yet she proposed (although abstract) a great method to assess DXF files. The pattern matching algorithm will work fine if the files element to be matched could be extracted properly at the first hand.

## **2.2 Automatic Online Assessment**

Trends towards teaching large numbers of students have led a number of researchers to examine ways in which the assessment of student assignments could be automated. In particular, courses on computer programming would seem a natural testing ground for computer-based marking. In fact, the origin of some study took place when computing still makes use of punch cards. Programming courses assignments generally take form in files of source code from a particular

programming language. These files need to be submitted to the course instructors to be compiled, reviewed and marked.

Similar to this process, Engineering Drawing course assignments are processed through the same method. Minus the compilation and testing, the drawing files of students are assessed by the instructors to check whether students have complied with the assignment instructions.

Previous studies on creating online programming assessment have produced a few notable conclusions. Although it seems to be different, but it is important to learn what these researches have produced in the process of creating automatic online assessment. The lessons learnt from those projects will help this research to guide itself to a direction where it will be able to satisfy the course instructors and participants.

Jackson (2000) in his work argues that difficulty lies in devising a foolproof approach to judging whether the outputs produced by that program are correct. The study argued that there are several aspects of assessment that can be performed much more accurately and efficiently by computer, and that can lead to more uniform marking across large groups of students. The study later presented that the fact remains, however, that there are still some aspects of assessment that are best performed by human tutors. The methodology proposed was conducting the testing (compilation) phase automatically, and then giving the tutor the opportunity to view the student's source code. In short, the study argues that the best way to perform an automatic assessment is combining the automated process (through software) and tutors talent. The arguments that Jackson presented is mainly because the nature of programming course assignment itself. The study believes that it's not enough the students' program are compiled and tested successfully. The study suggest that source code commenting remain as one of the essential marking factor in such courses. Jackson later elaborated how this aspect can be very subjective thus disallowing a fully automated approach to mark students' assignments.

Another study by Saikkonen, Malmi, and Korhonen (2001) suggests that a fully automatic assessment is indeed possible. The paper suggests that manual assessment



is fairly too much work for some tutors and believes that there are some things that need to be taken into consideration when creating exercises that are automatically assessed. First of all, exercises must be well-specified to the level of specifying an order for function arguments etc. These specifications are governed through a configuration file. There is a unique approach in this program. They believed that the marking shall focus on checking only the quality of the program. This eliminates the need to check students' comment on the source code. They resolve this by allowing students to retrieve almost immediately the result of their task and allow resubmission for each student. This way they make students learn and work on the quality of the program indeed and force them to forget the need of making comments on the code. After all, making comments on code can be quite subjective matter. The commenting process shall be designated to help future third party that is going to work with the same source code and not to be considered and the quality of the student's programming skill. This project can be seen as an important breakthrough as after it manages to help the instructors, it also forces the rearrangement of the course design (how the course is run). The program that is used by the author of the respective paper is claimed to work quite well and they have been able to assess all small exercises automatically. They also claim that in practice, the system has been stable and has required very little manual administration.

Both of the aforementioned study has presented two different points of view regarding online assessment. Both conducted study on programming based courses and have collected feedback from their research targets. Despite having a very different opinion on the proper way of conducting online assessment, they both showed that in order to make automated assessment fully function, a very detailed proper definition of the paper must be established in order to allow the automated functions work. This concept surely applies to the Engineering Drawing assignments which are built through tons of requirements. The second study presented that it's indeed impossible to make a perfect system without first adjusting the way that particular course is run. In this project the second study's approach will be exercised as this research main goal is indeed to create a fully automated system.

A better and detailed report on automatic assessment on programming course can be found in a research paper by Joy, Griffiths, and Boyatt (2005). They elaborate

splendidly in their study on their very BOSS online submission. It is well outlined in the study on few important features that must be contained by the online submission system. One of the most important approaches that are suggested is using a client-server model to build such system. The study also gives a few good suggestions on how the interfaces of such system should be. The consideration taken is the students demand for simplicity to use product to submit their work, both from the campus and when working at home, suggesting a web client as being appropriate. The system also highlights several marking aspects: correctness, style, authenticity. The authenticity parts lead them to develop a personalized assignment to each student. These way students have to work independently yet are forced to conduct discussion with their course mates. This practice is a better approach to allow students to engage in discussion instead of blindly copying others' work. The noteworthy point of this research that we admire is how this kind of project will help and shape student's character by forcing them to work ethically. Besides presenting the software itself, they also outline the result of students' acceptance of the software. We believe that such approach shall replicate in this project. However, we will only accommodate their findings once the core goals are achieved and there are extra time, cost, and resources available.

Another breakthrough was made by Rasila, Harjula, and Zenger (2007). Their work isn't related to programming like courses anymore. Instead they have developed a pilot project to automatically assess mathematic courses assignments. Interesting enough, their main intention is no longer to ease the burden of the instructors but instead to improve the quality of teaching in mathematics and related fields. There are several important points from their research. First mathematics assignment can be quite subjective; this is because students are also expected to use their creativity to solve problems. Now in order to allow automatic assignment, questions must be presented explicitly to avoid confusion. Second they also notice that students' grades have generally improved after the project implementation, suggesting that students are becoming more interested to solve mathematical problems after the existence of the system. In conclusion, this work showed that automatic assessment is adaptable to other type of learning. Besides presenting the automatic assessment feature, the project also contains extra feature of visualization of the problems. They have presented a survey at the end of their study that highlight the fact that student tend to

be motivated when working with visualized problems. This provides a motivation to the research that in Engineering Drawing field, similar approach should be feasible.

### **2.3 Engineering Drawing**

As explained in the first chapter of this report, Engineering Drawing is the form of technical drawing. It requires good understanding in engineering concepts and is used a communicator between engineers and their partners at work. Engineering Drawing is generally taught to engineering students on as early as their first semester. It will remain important and some engineers might eventually end up doing technical drawing for their whole career.

There are few important features that make it essential for engineering projects. Lines are the core element of engineering drawing. Lines will make up the other two core elements which are the Surface (including plane) and Edge. Surface detection and Edge detection will make up the core research activity in this project.

Despite the popularity of the usage of Computer Aided Design tools, some Engineers still prefer doing their drawing on the sketch using hands and some others prefer manual drawing tools. This process is coherent in the academic world where some students have to have to make their drawings hands on for the earlier stage of their study.

There are several unique characteristics that make Engineering Drawing differ in nature with any other drawing. In Engineering Drawing, every design is drawn precisely and concisely. It is usually made to scale with the real object. There are two “languages” used in the process. The Graphics language describes the shape whereas the Word language describes the exact size, location, and specification.

Engineering Drawing is governed by certain international standard. The implementation of the standard might differ on style but it still intends to allow the drawings to convey the same meaning to everyone who reads them. The first standard is the drawing sheet standard which simply refers to the size of the paper that contains the drawing. It can also translate into the definition of margin,

orientation of the paper. There is also a standard that governs the scaling of the drawing. In Engineering Drawing, scale is defined as ratio between the linear dimension of a drawn representation of an object and the actual object. Some objects in the drawing that might be independent to the scaling must be accompanied with a dimension note that states the actual size of the object. Beside the standard on the drawing, there is also a standard that rules over the lettering in the drawing. Letters in the drawings must have at least two properties which are uniformity (consistency of letters) and legibility (position of letters). In CAD, this standard isn't much of a concern. While in manual drawing, this standard is a bit tedious as there are a lot rules that has to be followed. The last standard is the Line Types which might differ on different kind of industry.

## 2.4 DXF (Drawing Exchange Format) File Type

According to official DXF 2012 documentation, DXF (Drawing exchange Format) is defined as graphics exchange files, it is a kind of sequential file, and also is a kind of data file which is including entity commands and geometric data under a certain symbol regulations of block code. DXF, which is including all the information of DWG file, is able to form figures faster. It is of good data mutual reading with third-party files, and is easily and directly read by much software such as 3DMax, MatLab and so on, but as development software. The DXF file is a text file, when opened through a text editor, the file will show the tagged representation of data contained.

The DXF format is a tagged data representation of all the information contained in an AutoCAD drawing file. *Tagged data* means that each data element in the file is preceded by an integer number that is called a *group code*. A group code's value indicates what type of data element follows. This value also indicates the meaning of a data element for a given object (or record) type. Virtually all user-specified information in a drawing file can be represented in DXF format.

In the DXF format, the definition of objects differs from entities: objects have no graphical representation and entities do. For example, dictionaries are objects, and not entities. Entities are also referred to as *graphical objects* while objects are referred to as *non-graphical objects*. Entities appear in both the BLOCK and

ENTITIES sections of the DXF file. The use of group codes in the two sections is identical. Some group codes that define an entity always appear; others are optional and appear only if their values differ from the defaults.

The HEADER section of a DXF file contains the settings of variables associated with the drawing. Each variable is specified by a 9 group code giving the variable's name, followed by groups that supply the variable's value.

The typical elements in the ENTITIES SECTION are as follows.

- Point: Point is the smallest component of the elements. And its coordinates are the location information.
- Line: Linear positioning information is determined by the two endpoints.
- Circle: the center point and the radius identify Circle location information.
- Arc: The Location information is composed of arc heart, radius, and the start and termination arc angles.
- Curve: Curve is determined by a series of points, lines and arcs.
- Text: The description mainly includes text contents, text positioning and font sizes and so on

## **2.5 Differential Conversion from DXF into SVG and Vice Versa**

Ahmad and Haque (2001) proposed a breakthrough in reading DXF files through their work. Their initial intention was creating a bridging interface between designers and manufacturers through feature recognition. Important achievement on their paper is that they have successfully implemented such task. They outline the detail structure of DXF file and group them into categories. Their job will make great reference on how to deal with DXF files. However they don't propose any method to convert DXF files into other data format.

Now a proper bidirectional conversion details from DXF to SVG is outlined splendidly by Fahiem and Farhan (2007). They emphasize on the reality that there are many algorithms available to do such task. They basically work through extracting the boundary of the image thus converting them into respective SVG

elements. They also outline the fact that SVG is faster to be processed in the web environment. They justify this by presenting several good statistics.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

This chapter is about the research processes and activities in this project. In a nutshell, the research can be broken down into four layers of activities. These layers are interdependent, timely fashioned, and fairly manageable. They are extracted from the basic software prototyping methodology.

Software prototyping is generally viewed as the best software engineering approach to develop a web based system. The main factor is because web based system is hard to be visualized and the flow of data or information might seem confusing to some novice users and developers. Prototyping allow target users to incrementally view and observe the progress of the project.

Prototyping generally starts by acquiring software requirements or specification by interviewing the targeted users. Developers then extract the core of the requirements to generate an initial prototype that serve the main features or functionality expected from the software. The users are then invited to view and criticize the prototype. The developers later gather feedback from the users to enhance the prototype. This process is done repeatedly until every single original features and users' feedback is adapted into the software.

In this project, the implementation of the software prototyping methodology is given into this 4 layer of activities:

- Requirement gathering
- Conversion algorithm development
- Assessment algorithm development

- Iteration & final delivery

Prototyping is unique as it enforces iteration after each software delivery if the partial product is rejected by the stakeholders. There are some benefits resulted from this approach. First, developers can obtain feedback from users in the formative stages of the project. This method also allows developers to gradually check their work against software original requirements. Developers can also perform regular assessment on whether the project is able to meet the timeline and budget constrain. Finally, software prototyping avoid risks of changing product requirements unexpectedly. A high level of user involvement ensures that every single user's expectation is well captured in the whole software development.

Prototyping is the best option to be used for a system composed of modular packages. This is because during the testing period some packages might need to be modified or taken out where few should remain. This way the working module can remain safe while the problematic modules are fixed. With a good and efficient software design, developers can develop their software package to package without compromising the integrity of the software as a whole. The application of this module/package based development in this project is elaborated in section 3.4 & 3.5.

### **3.2 Research Activities**

There are two essential research questions laid down upon the birth of this project. The first question is the quest to convert DXF file into SVG. The DXF format is actually manageable as it contains ASCII characters, however it has versions issue. DXF 2007 file is designed in a much different way compared to DXF 2009 file for example. There are a lot of variations on the DXF format as usually a new version is published in conjunction with the launching of new version of AutoCAD. The conversion algorithm must be able to somehow understand the variation and format of DXF and convert its data into respective SVG. Details on conversion process in further elaborated in section 3.4. The conversion module of the software will be the first module to be prioritized in the prototyping iteration.



The next research problem remains to be solved is referring to the process of marking the assessment files. The newly produced SVG files must be processed in a way that there should be no bias in marking them. This means that the very least, the algorithm shall be able to check the correctness or the accuracy of the students' answers.

After acquiring the specification from the stakeholders, their requirements can be directly implemented into the software prototype. It is not expected however that the original specification to completely fit the stakeholders' expectation therefore the iteration mechanism is the key to incrementally increase the capability of the algorithm.

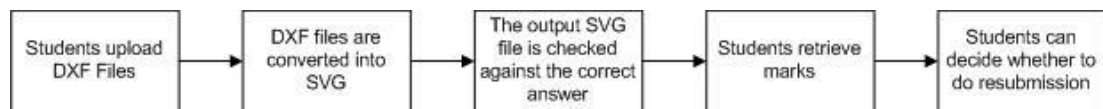


Figure 1 - How the Software Shall Work

The two research activities shall be the foundation of how the software works as a whole. The software will roughly work in this following manner:

- Student upload their assignment file into the website
- If application is not processing other files then the file shall be directly converted
- The result of the conversion is later process through the assessment algorithm
- Right after being assessed, the software shall calculate the mark of the assignment and directly display or alert the student about it
- The student can decide whether to perform resubmission after looking at his/her result

The whole process is further illustrated in Figure 1 (next page).

### 3.3 Requirement Gathering

The first layer will take form in a requirement gathering activity. In this pre-development study stage, we will create a survey that is based on the literature of automatic online programming assessment. Using the survey we will measure the

applicability of the approaches discussed in those researches in engineering drawing context. The objective for this study to clear any assumptions made during the initiation of this project in order to get clearer understanding on the research subject.

We hope to understand specifically how the entire courses being run. We have done pre-research study to determine that each department has their own style of conducting the course. All of those courses actually have the same objective, but since their discipline apply Engineering Drawing differently, several adjustments are then necessary. This study will be conducted in two approaches. First is a survey targeting the participants, and second is the interview with the course instructors. We won't touch every single details of the proposed study in this report but we will highlight the important ones.

In the survey, we expect to cover on several technical or non-technical issues related to the course. First we need to determine exactly how many assignments are generated every semester. We will also measure the statistic of total participants of the course and teacher-students ratio each semester. Apart from that we will also collect several files or assignment created by previous students in order to be used as testing sample. We will also observe how the end to end process marking has been done all this time and obtain a marking pattern that will be implemented in the algorithm. We will also ask both parties on their expectations of this project. We will also explore on their alternative preferences of our proposed solutions. We will expect some necessary adjustments later on if there are really great inputs from them. Finally after this gathering is done, we will measure the gathered data against our objectives and limitations. If things don't work out well, we don't mind to extend the period of study.

### **3.4 Converter Module Development**

It was mentioned many times in earlier chapters that the end target of this research is to produce software. Now good software has to meet few criteria. One of the most important criteria is how effective the development process of the software. We will elaborate on this factor more on this report whereas the other criteria will have to wait until the requirement gathering is done.

The emergence of object oriented programming changes the way software is built. While there are other better ways to define the term, the quickest approach to understand object oriented concept is by realizing that everything is an object with its own properties and behaviours. Everything includes files, hardware, & data. In the relevancy to the programming concept, the representation simply means how to write down pieces of code to logically present the real or physical identity of each object.

| Sample of DXF ASCII Characters  | Sample of SVG Tags   |
|---|--|
| <pre> 0 SECTION   2 HEADER   9 \$ACADVER   1 AC1015   9 \$ACADMAINTVER   70     6   9 \$DWGCODEPAGE   3 ANSI_1252   9 \$INSBASE   10 0.0   20 0.0   30 0.0   9 \$EXTMIN   10 0.0 </pre> | <pre> - sodipodi:docname="Yes_check.svg" sodipodi:docbase="/home/fizyk/anizo/wiki/SVG/niemoje"&gt; &lt;metadata   id="metadata1375"&gt;   &lt;rdf:RDF&gt;     &lt;cc:Work       rdf:about=""&gt;       &lt;dc:format&gt;image/svg+xml&lt;/dc:format&gt;       &lt;dc:type         rdf:resource="http://purl.org/dc/dcmitype/StillImage" /&gt;     &lt;/cc:Work&gt;   &lt;/rdf:RDF&gt; &lt;/metadata&gt; &lt;defs   id="defs1373"&gt;   &lt;linearGradient     id="linearGradient2250"&gt;     &lt;stop       style="stop-color:#008700;stop-opacity:1;" </pre> |

Table 1 – Sample of DXF & SVG File

In this research as outlined earlier one of the functionality of the software is to convert DXF files into the equivalent SVG versions. Now before we discuss about the conversion, we must first be able to plan strategically how to ease our future tasks. We expect that there will be a lot of codes to be written repeatedly. We also expect to update and improve our algorithm over time without having to changes the previously implemented version unless necessary. All of these issues are classical and have been proved to be effectively resolved by the object oriented programming approach. Table 1 shall illustrate clearly how each file type is interpreted by a standard text editor.

### **3.5 Assessment Module Development**

Once the original files are converted into web friendly format, it's time to move on to the development of marking algorithm. The main idea of marking is to have the correct version of the assignment stored in the server and having each student's submissions checked against it. There might be some issues with consistency of the algorithm as we can't really ensure that computerized process can understand and distinct the students' creativity in crafting ups their assessment. The problem doesn't lie on the conversion process but on the checking phase instead. We propose the different counter measures.

First we have to alter the standard conversion process so that we will ensure that the conversion process is coherent with the correct answer. This way is good in easing the development burden, but will cost some students. This will happen especially if there is not clear elaboration on problems. Different students might interpret differently. Even after so much effort to clearly define the problems, there are still possibilities that sometimes students can't follow exactly the way the assignment is crafted. There is also a possibility to allow marking process to be done fast enough so that student can retrieve feedback immediately and do resubmission. However, it's best to wait until the pre-development study results are gathered to clear all of these assumptions.

The next issue to be taken care of is the problem of checking the originality of the students' assignments. By allowing submission electronically, we basically open up a possibility of easier plagiarism and cheating. Some will suggest the use of personalized assignment in order to counter this issue. However this idea will be very hard to be implemented in Engineering Drawing. It is important to remember that Engineering Drawing is done precisely and concisely. The best approach to create personalized assignments is by asking the tutors to create them. This might be infeasible when there is a very big ratio between instructors and participants. Another possible way is to generate the algorithm to automatically create multiple personalized assignments from a single source of file. This way is possible but there has to be some algorithm implemented to ensure that each generated outcome match the international standard.

The last but not least is still about the accuracy of the submissions. This is where SVG will come in handy than original DXF. They will generally have lesser size and are consequently processed faster. The reading will utilize XQUERY and XPATH. Both will be utilized to go through each of the file's tags and having its attributes compared with the counterpart of the original version. We expect to take some time before finally deliver the best algorithm to cater this issue.

### **3.6 Delivery, Iteration, & Post Development**

Once core modules are ready, they will be embedded into JSP based application and therefore ready to be implemented. We expect to have very few challenges in this phase. JSP allows overnight development; this has allowed us to focus all of our efforts on the creation of necessary classes which were explained earlier.

Once the first batch of delivery is ready, the stakeholders will be invited to test out the system. There will be post development survey conducted during this stage in order to challenge the relevancy of the project with the objectives. This survey will be consistent and irreplaceable even after the rejection of the software and the delivery must be repeated.

### **3.7 Tools Required**

| No | Name                  | Roles                              |
|----|-----------------------|------------------------------------|
| 1  | C#                    | Core programming language          |
| 2  | .NET                  | Development platform               |
| 3  | ASP.NET               | Web development framework          |
| 4  | Visual Studio 2010    | Integrated development environment |
| 5  | AutoDesk AutoCAD 2012 | DXF default reader                 |
| 6  | W3C- SVG              | Alternative of DWG & DXF           |
| 7  | W3C - XPath, XQuery   | Tools for reading through the SVG  |

### 3.8 Gantt Chart

| Task                          | Jun | Jul | Aug | Sept | Oct | Nov | Dec |
|-------------------------------|-----|-----|-----|------|-----|-----|-----|
| Literature Review             |     |     |     |      |     |     |     |
| Initiate project proposal     |     |     |     |      |     |     |     |
| Requirement Gathering         |     |     |     |      |     |     |     |
| Converter Module Development  |     |     |     |      |     |     |     |
| Data Analysis                 |     |     |     |      |     |     |     |
| Assessment Module Development |     |     |     |      |     |     |     |
| First Delivery                |     |     |     |      |     |     |     |
| Post-Development Study        |     |     |     |      |     |     |     |
| Iteration                     |     |     |     |      |     |     |     |
| Final Delivery                |     |     |     |      |     |     |     |

Table 2 - Project Gantt Chart

### 3.9 Key Milestone

Up until this point of development, this research has produced several significant results. The research has completed at least 60% of its activities. The research also manages to conduct a survey that was welcomed with good enthusiasm from targeted stakeholders. The survey results have been used to check assumptions made in the earlier phase of the project. There has been progress also in the software development as a whole. This project has produced development framework for the aforementioned software along with its possible features and limitations. Details over these achievements are available on the next chapter.

## CHAPTER 4

### RESULTS AND DISCUSSIONS

#### 4.1 Pre-Development Survey Results

We make a few assumptions at the initiation phase of this project. Those assumptions are reflected into few key decisions done that will decide this project. We will utilize the results of this survey to support or justify those decisions.

The survey is composed of 11 questions and there were 78 respondents. It was disseminated only to Universiti Teknologi PETRONAS students through online media. The respondents filled up the form starting from 19 July 2012 – 24 July 2012.



Figure 2 – Question 1

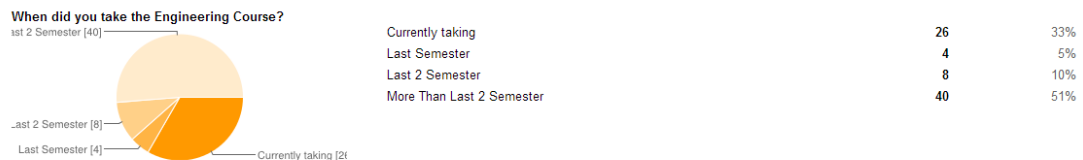


Figure 2 – Question 2

The first two questions are meant to determine the demographics of respondent. This is to ensure that people who are responding to this survey have taken up the Engineering Course or are currently taking it. This way, we ensure that their responses are accurate and representative for this project.

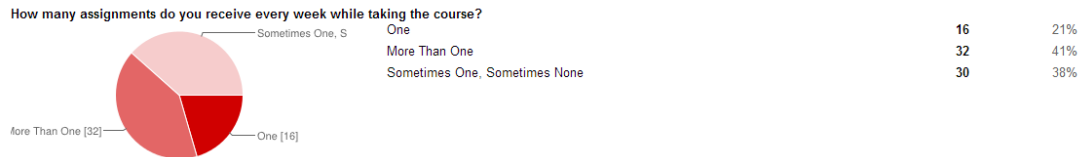


Figure 4 – Question 3

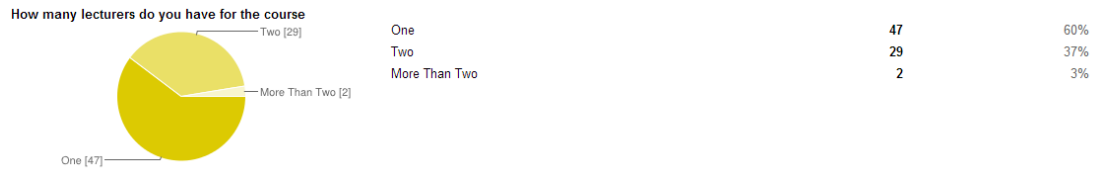


Figure 5 – Question 4



Figure 6 – Question 5

The next three questions are created to find out the proportion of instructors and participants of the course. If we take the majority responses in this survey each of 100 students taking Engineering Drawing course have to submit approximately 2 assignments every week to be later assessed by a tutor and a lecturer. Since UTP is offering 3 different set of the course the total of students shall be about 300 and the number of lecturers (and tutors) shall be around 6. If the students are obliged to submit their assignments for 10 weeks per semester, this means that the 6 instructors have to mark around 6000 assignments per semester. Assuming each instructor is assigned equal responsibility, each of them is responsible for 1000 assignments per semester.

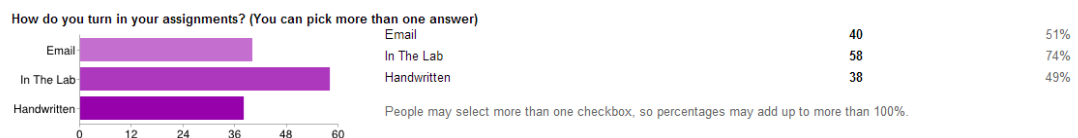


Figure 7 – Question 6



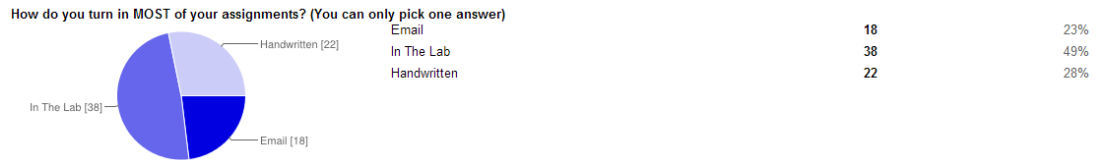


Figure 8 – Question 7

The next two questions are set to find out the submission mechanism of the assignments. The two questions are asking the same thing. However, in question 6, respondents are able to choose more than one submission methods. In question 7 on the other hand, respondents can only pick one of them. Both questions however indicate that submissions happen most of the time at the lab.



Figure 9 – Question 8

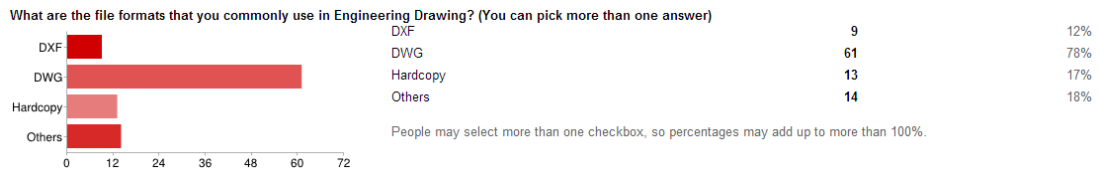


Figure 10 – Question 9

Question number 8 and 9 is meant for finding out the software and tools utilized at Engineering Drawing course at UTP. Question 8 indicates that AutoCAD is the major tool used in this course. This is in line with the original assumptions. However question 9 shows that DXF is not as popular as the other file format. The only justification of DXF usage in this research is because its simplicity as explained in Chapter 2. The fact that it's unpopular can be catered easily as all AutoCAD versions support this format (despite the fact that it's rarely used). Another implication is that target users must also be informed that this software can only work with DXF files.

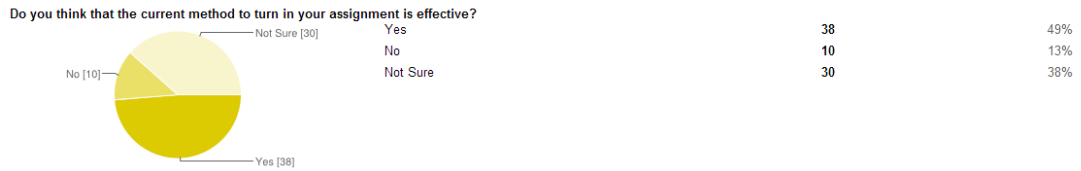


Figure 11 – Question 10

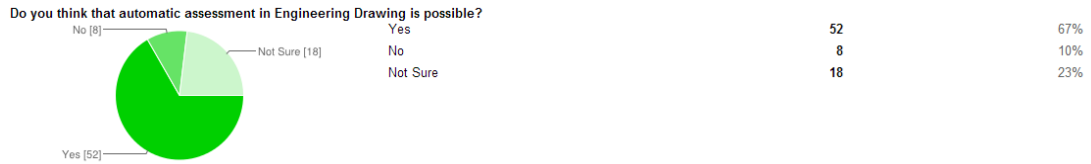


Figure 12 – Question 11

The last two questions examine the satisfactory level of the course participants towards the traditional marking method. A positive majority response is expected as the fact remains that the burden to mark is left to the instructors. However the respondents believe that automation in the assessment process is possible as indicated by question 10.

## 4.2 Converter Module Prototype

It will take some time to come up with a set of C# classess capable to transform all set of data stored in DXF format to its SVG counterparts. However, we managed to create several classess capable to transform some of the whole DXF data library. We are going to demonstrate how to use one of the classess to convert a DXF drawing illustrating a circle into SVG. Figure 4 shall represent how the Circle should look like.

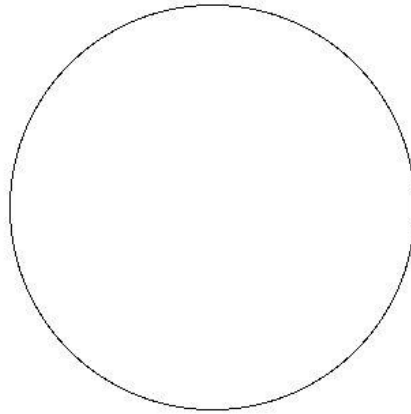


Figure 13 - A Simple & Plain Circle

This circle despite its simplicity still takes up 7000 lines to be represented in DXF format. However the only significant data is store in the ENTITIES section. The details are further illustrated in Figure 5.

```
...  
SECTION  
2  
ENTITIES  
0  
CIRCLE  
5  
188  
330  
1F  
100  
AcDbEntity  
8  
0  
100  
AcDbCircle  
10  
9.667172840374078  
20  
13.48236319755094  
30  
0.0  
40  
2.702626290815926  
0  
ENDSEC  
...
```

Figure 14 - A Simple & Plain Circle Represented in DXF

The above code has some important meanings. The figure below “10” represents x-axis coordinate of the circle midpoint, where the figure below “20” represents the y-axis coordinate. Finally, the figure below “40” represents the radius of the circle. Knowing these properties in DXF and how they are being represented in SVG shall give us knowledge on how to build a program to convert this circle from its DXF format into SVG. Figure 6 illustrates a flowchart on the algorithm of the program.

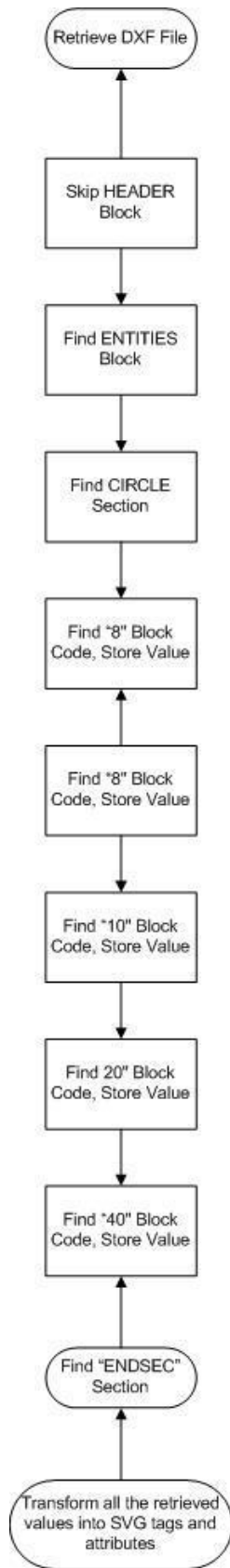


Figure 15 - Circle DXF to SVG Conversion Algorithm

After conversion, the resulted SVG file will look like what is represented in Figure 16.

```
<svg xmlns="http://www.w3.org/2000/svg"
version="1.1">
<circle cx="9.667172840374078"
cy="13.48236319755094" r="2.702626290815926"
stroke="black" stroke-width="1"/>
</svg>
```

Figure 16 - SVG Resulted from the Conversion

### 4.3 Marking Module Development

Once we are able to convert the DXF files into SVG, we can implement an XPath algorithm to traverse onto both the student’s submission and the correct answer prepared by the tutor. The algorithm shall traverse onto each of the generated SVG tag of the student’s file and examine its attributes; the value will later be compared to the value of the similar tag from the student’s answer. This way we can directly know whether the particular part of the assignment has been done correctly. For example in Table 3 we show two different SVG representation of a circle. One is meant to come from the tutor where as one is from the student. The student apparently made some mistake as he didn’t give the correct data on his circle radius. The algorithm shall detect this difference given the data is stored in the same SVG group tag.

| Student  | Tutor  |
|--|--|
| <pre>&lt;svg xmlns="http://www.w3.org/2000/ svg" version="1.1"&gt; &lt;circle cx="9.667172840374078" cy="13.48236319755094" r="2.702626290815926" stroke="black" stroke- width="1"/&gt; &lt;/svg&gt;</pre> | <pre>&lt;svg xmlns="http://www.w3.org/2000/ svg" version="1.1"&gt; &lt;circle cx="8.667172840374078" cy="13.48236319755094" r="2.702626290815926" stroke="black" stroke- width="1"/&gt; &lt;/svg&gt;</pre> |

Table 3 – Comparison between Tutor’s Circle and Student’s Circle

## 4.4 Prototype Screenshots

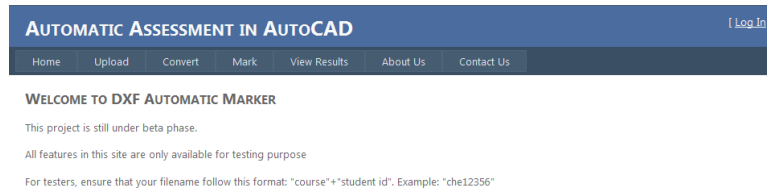


Figure 16 – Homepage screenshot

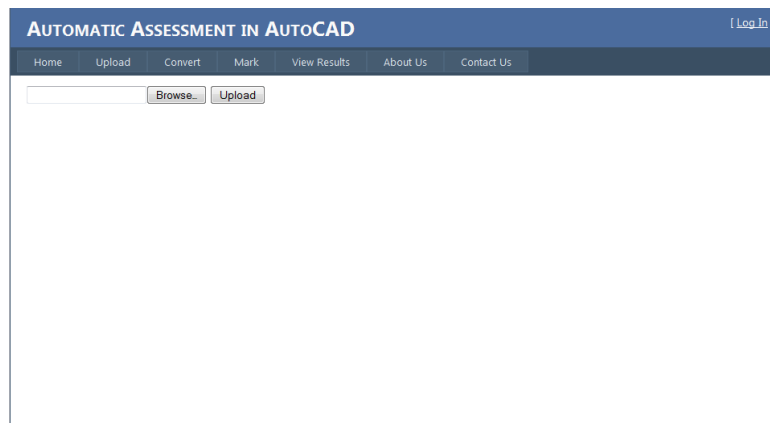


Figure 17– Files uploader screenshot

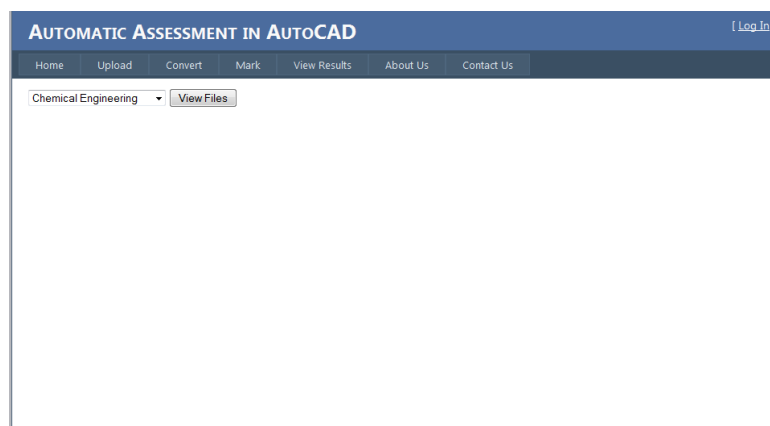
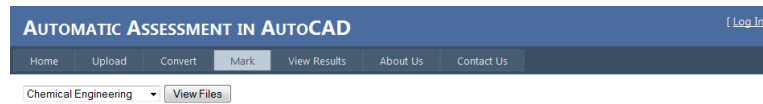


Figure 18– Converter screenshot



---

Figure 19– Marker screenshot

## 4.5 Conclusion

The survey manages to justify some assumptions made in the initiation phase of the project. The unjustifiable assumptions are also highlighted in order to help the researchers anticipate them. Overall the survey is proved to be beneficial to the research. The survey also managed to attract a significant number of respondents (n=79) as compared to other Final Year Project surveys.

The successful creation of an initial prototype of a converter module can be said to be remarkable. The module, despite seeming difficult, is actually composed of simple programming that is replicable with other objects.

While the complete early prototype of the assessment module is yet to be produced, the idea of how to create it has emerged in this research. With the given time, the code needed for the module can soon be tested.

Overall, the progress of this project has been very promising. Obvious needed features such as GUI are not yet discussed, but will be due for the next reporting. The details on where this project shall go after this phase are discussed in the next chapter.



## CHAPTER 5

### CONCLUSIONS & RECOMMENDATIONS

#### 5.1 Conclusion

##### 5.1.1 Relevancy of Objectives & Scope of Study

This project has very clear objectives since its inception. At the end of this study, this project is expected to produce software that at least has two capabilities. First it must be able to allow Engineering Drawing students to submit their assignment files conveniently. Currently they have to either hand in the files in the lab or emailing them to the tutors/lecturers. The second is performing automatic assessment into the files. Extra features that might be added later on to this software will be adjusted with the providence of available cost, time, and resources.

Since there are a lot of ways and tools to do Engineering Drawing, the idea of creating software capable of automatically assessing students' assignments in that course might seem to be an ambiguous idea when the scope of the study is not defined properly. This is because without carefully selecting the scope of the research, the project might become too big and infeasible. Therefore, in this research the scope has been carefully limited since the beginning of this project.

AutoCAD was chosen as the companion tool in this research as it is used in Universiti Teknologi PETRONAS (as proven by survey results). This will allow the researchers of this project closely measure their research against reality that is happening in a real Engineering Drawing course. AutoCAD is also used widely either in the industry or academic world. In fact, it has been the industry standard.

That option has allowed this project to acquire two scope of study in one move. The first one is technological scope that is to limit this research to focus on AutoCAD software and its file format offspring: DXF. The second scope is the non-technical scope where this project manages to obtain potential stakeholders and software testers which are UTP Engineering Drawing students and lecturers.

### **5.1.2 Relevancy of the Literature Review & Methodology**

This project is considered to be unique as there has been no similar research done previously. This comes with consequences that it is fairly hard to obtain relevant literatures that can be used to support this research. However this project managed to clearly define the abstraction of its goals early on and therefore is able to determine the criteria of the relevant literatures.

As this project is fairly unique it is important to set a methodology that can sustain time and budget limit. This is the reason on choosing a software prototyping methodology. Software prototyping ensures a delivery of core feature gradually but not sacrificing the extendibility of the main modules. This is to ensure that the initial prototype is able to perform conversion of DXF into SVG then afterwards it will be enhanced to be able to mark the produced SVG. After able to perform the core feature extra functionalities will added that shall accomplish the software to be reliable and usable.

### **5.1.3 Progress & Next Step**

As elaborated in Chapter 4, we have developed an initial framework capable to convert DXF file displaying a circle object into SVG. We also demonstrate on how assessment shall be conducted to that file. The next step is now extending this framework in order to be capable to work with other objects or shapes used in the engineering drawing context. The software has to be able to work with other shapes like rectangle, oval, or polygons. We just need to ensure the integrity of the algorithm by testing it extensively.

## **5.2 Recommendation**

To be able to create automatic assessment software that is able to work with multiple tools used in Engineering Drawing, one must first acquire a lot of resources. In this study we are only able to cover file formats generated by AutoCAD software due to our limited resources.

The question now is not whether development of software capable to fully cater all file formats generated by major Engineering Drawing tools is necessary. The question is actually what other CAD tools that has to be prioritized beside AutoCAD.

The decision to perform conversion on the file format into an open one such as SVG might come in handy in extending this project. We can have one marking algorithm for all kind of different file types. We just need to sure multiple conversion algorithm is deployed for different file formats.

Some popular CAD tools that should be prioritized include the CATIA, SolidWorks & Ansys. Each has their own file format that is professionally documented. With properly assigned resources, creation of software capable to perform automatic assessment with all kind of Engineering Drawing software is certainly highly possible.

## REFERENCES

- [1] Ahmad, Nafis & Haque, Anwarul A.F.M (2001). Manufacturing Feature Recognition of Parts Using DXF File. International Conference on Mechanical Engineering, 4<sup>th</sup>, 111-115.
- [2] Fahiem, Muhammad Abuzar & Farhan, Saima (2007). Representation of Engineering Drawings in SVG and DXF for Information Interexchange. International Conference on Circuits, Systems, Electronics, Control & Signal Processing, 6<sup>th</sup>, 592-596.
- [3] Jackson, David (2000). A Semi-Automated Approach to Online Assessment. Innovation and Technology in Computer Science Education, 6<sup>th</sup>, 164-167.
- [4] Mike Joy et al (2005). Automated The BOSS Online Submission and Assessment System. ACM Journal on Educational Resources in Computing, Vol. 5, No. 3, September 2005. Article 2.
- [5] Othman, Nurliyana (2007). Automated AutoCAD Drawing Assessment System For Engineering Graphics Course. Seri Iskandar: Universiti Teknologi PETRONAS.
- [6] Rasila, Antti et al (2007). Automatic assessment of mathematics exercises: Experiences and future prospects.
- [7] Saikkonen, Riku et al (2001). Fully Automatic Assessment of Programming Exercises. Innovation and Technology in Computer Science Education, 7<sup>th</sup>, 133-136.