

Step File Converter Application Research Document

Alan Halpin - Institute of Technology Carlow

Started 14/10/2020



Supervisor: Joseph Kehoe

Student Number: C00229361

Table Of Contents

Abstract	2
Introduction	2
Objectives	2
Background	3
CNC Machining (3D Printing)	3
CNC Programming	3
Machines	4
STEP	5
What is a STEP File?	5
Definition of a STEP File	5
STEP Application Protocols (AP)	7
Express	7
STEP Data Structure	8
XML	9
Euler Bernoulli Beam Theory	10
Existing Technologies	11
Technologies	12
C++	12
C#	12
Conclusion	13
API	14
Solid works API	14
Conclusion	16
Plagiarism Declaration	17
Declaration	17
References	18

Abstract

This project aims to build a tool to aid in 3D CNC printing. The problem is: when carving a piece from a solid block, an engineer needs to manually determine the best order in which to carve out each face so the piece remains stable throughout the process. If the piece does not remain stable, it may have imperfections. This tool is a proof of concept. This document is intended to provide details required to begin developing this project, what technologies are suitable and information on STEP files and how this problem may be solved.

Introduction

The STEP File Converter is a tool for CNC engineers, designed to automate a lengthy and costly process. Nothing like this application currently exists. The value is derived from the process being automated. This process involves performing vibration analysis on different permutations of an object until a satisfactory sequence of faces is found. Vibration analysis is performed to ensure the piece worked on remains stable throughout the cutting process. The proposed tool will extract geometrical data from a STEP file and catalogue the different faces of the object described. One by one these faces will be subtracted from a solid block shape. Vibration analysis is performed between each step, if the piece does not pass the requirements then a different face will be used. This will eventually create a list of faces to cut in order. Ex. A,B,F,G,C might be an order in which to carve first. Although vibration analysis already exists within various CAD applications, the process described only exists in a manual implementation. This tool will use the Solid Works API to outsource the calculations and identify the optimal cutting order within the application. An engineer will pass a STEP file to the program and the program will start working on the results. Depending on the parts complexity this process may take some time. Typically this process is done manually and can take up to three weeks to complete, this tool hopes to automate that process and reduce that time down to one day.

Objectives

1. Read a STEP file and extract data.
2. Catalogue faces and features.
3. Procedurally apply the Euler formula to the shape to determine the best order in which to carve out each face so the shape and machine tool remain steady throughout the entire process.

Background

CNC Machining (3D Printing)

Computer Numerical Control (CNC) is the automated control of machine tools and 3D printers by a computer.

CNC machining surpasses manual machining by an operator as it is near 100% accurate, no human error and it is much faster. A CNC machine's measurements start at 1 thousandth of an inch, it is incredibly accurate. Although the true value comes from the software, being able to translate designs from CAD software; Solidworks, TurboCAD, etc.. to the CNC machine in order to produce your piece.

CNC Programming

The machines are operated via numerical control, the software written to do this is known as G-code. G-code was first developed in the 1950's by MIT, to implement early computer aided machining. This language has developed along with the technology used for machining.

CNC Programming makes it possible to pre-program the speed and location of the tools and when to change tools all automatically for an assembly line without any human intervention.

Before a piece is carved an Engineer often simulates the machining to check for errors in the code or determine the best order in which to carve the faces so the piece and cutting tools stay stable throughout the process. (Astro Machine Works, 2017)

Machines

There are several Different types of CNC machines;

- Mills
- Lathes
- Plasma Cutters
- Electric Discharge Machines

Mills

Milling is the cutting and drilling of material. Milling machines use a rotating cylindrical tool called a milling cutter that is held in a spindle.

Milling machines can cut in different angles and move along several axes, depending on the machine they can have between 2-5 axes.

Lathes

A lathe is a machine tool that spins the workpiece rapidly about an axis to perform several operations; cutting, sanding, knurling, drilling, deformation, facing and turning. As the piece is spinning, the tools are held in an arm and fed into the piece. The nature of the lathe produces an object with symmetry.

Plasma Cutters

A plasma cutter is a machine that works by sending an electric arc through a gas that is passing through a constricted opening. This gas may be nitrogen, argon or oxygen. This causes the gas to get so hot that it transforms into plasma. CNC has allowed plasma cutters to automate the shape production process. (Torchmate, 2020)

Electric Discharge Machines

Electric discharge machines work by using electrical discharges. The material is removed from the piece by many discharges occurring between two electrodes that are separated by a dielectric liquid under an electric voltage.

STEP

What is a STEP File?

Standard for Exchange of Product model data (STEP) is a file format used to store 3D modelling data that is considered neutral. The format is considered neutral as it contains a complete description of the product which is independent of any software, OS or computer system. The file uses plain text to portray this information, they chose to do this so it is both machine and human readable. These files are written in EXPRESS, a data modelling language.

ISO 10303-21 is used to store data that represents 3D models in CAD and manufacturing information. These files are written using an ASCII structure aka Plain Text, by nature this makes them easy to read. Having one instance (or step) per line in the format #1, #2, #3 etc... These steps can be located in the DATA section. (Datakit, 2020)

Definition of a STEP File

A step file consists of two parts;

- HEADER section
- DATA section

HEADER

- FILE_DESCRIPTION
- FILE_NAME
- FILE_SCHEMA

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP203' ),
  '1' );
FILE_NAME ('Cube.STEP',
  '2020-10-26T15:18:52',
  ( '' ),
  ( '' ),
  'SwSTEP 2.0',
  'SolidWorks 2020',
  '' );
FILE_SCHEMA (( 'CONFIG_CONTROL_DESIGN' ));
ENDSEC;
```

Figure 1: HEADER Section of a STEP file

The **File Description** may define which application protocol is being used, the version and conformance option of the file.

The version may be;

- '1' for the original standard 1994
- '2' for the technical corrigendum 1995
- '3' for second edition

The conformance may be;

- '1' for internal
- '2' for external mapping

The **File Name** contains the name of the part, a time stamp and it may also contain the name of the software used to create the file.

The **File Schema** may vary as there are many different schemas defined in the Express language.

The schema determines how the file should be read and how the data can be extracted. If the version of the file is '3' then there may be multiple schemas used.

DATA

```
DATA;
#1 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #94, #177, ( #134 ) ) ;
#2 = LINE ( 'NONE', #93, #145 ) ;
#3 = DIRECTION ( 'NONE', ( -1.000000000000000000, -0.000000000000000000, -0.000000000000000000 ) ) ;
#4 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, -1.000000000000000000 ) ) ;
#5 = DATE_AND_TIME ( #217, #92 ) ;
#6 = EDGE_CURVE ( 'NONE', #141, #85, #34, .T. ) ;
#7 = CARTESIAN_POINT ( 'NONE', ( 100.0000000000000000, 100.0000000000000000, 0.000000000000000000 ) ) ;
#8 = ORIENTED_EDGE ( 'NONE', *, *, #65, .F. ) ;
#9 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, 0.000000000000000000 ) ) ;
#10 = PERSON_AND_ORGANIZATION ( #147, #156 ) ;
#11 = PLANE ( 'NONE', #105 ) ;
#12 = APPROVAL_STATUS ( 'not_yet_approved' ) ;
#13 = APPROVAL_PERSON_ORGANIZATION ( #102, #197, #168 ) ;
#14 = ORIENTED_EDGE ( 'NONE', *, *, #158, .F. ) ;
#15 = VERTEX_POINT ( 'NONE', #51 ) ;
#16 = EDGE_CURVE ( 'NONE', #26, #85, #149, .T. ) ;
#17 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, -0.000000000000000000 ) ) ;
#18 = ADVANCED_FACE ( 'NONE', ( #108 ), #181, .F. ) ;
```

Figure 2: DATA section of a STEP file.

There may not be any order to sequence of commands here “it is not possible to sequentially read a STEP file. Entities can be in any order and can reference other entities forwards and backwards in the file. Therefore the entire file has to be read into memory and tokenized before parsing.” (Wiki, ISO 10303-21, 2020) The data needs to be processed in order to understand which values belong to each face of the object. The data section is defined by a series of instances or ‘steps’ that define that object. The information contained is a collection of features and faces.

STEP Application Protocols (AP)

The Application Protocol can be identified under FILE_DESCRIPTION in the example above (Figure 1). There are many different AP's and they are used to define a data exchange standard for various products at a certain stage in a product's life cycle. Some AP's are specialised for a certain family of products such as AP-210 which is for printed circuit boards. The most commonly used protocols for CAD are; AP 203/214, 224, 240 and 238. These protocols define the complete design and manufacturing process of parts.

Application Protocols may define similar information. Data is shared among these protocols by mapping each protocol model into a set of integrated resources (IR) to make the Application Interpreted Model (AIM). The AIM allows data sharing as it uses the same definitions and concepts across all the different protocols. (STEP Tools, Inc. 2020)

Express

STEP files are written in an object oriented data modelling language known as Express. Express was designed to represent product data through a schema. The schema is written to define what a set of data should conform to in order to be validated. A schema can be seen as a set of rules that the data section must follow. This allows the information to be parsable by humans and computers as these schemas are defined and have rules set. EXPRESS is similar to XML in structure (tree structure).

STEP Data Structure

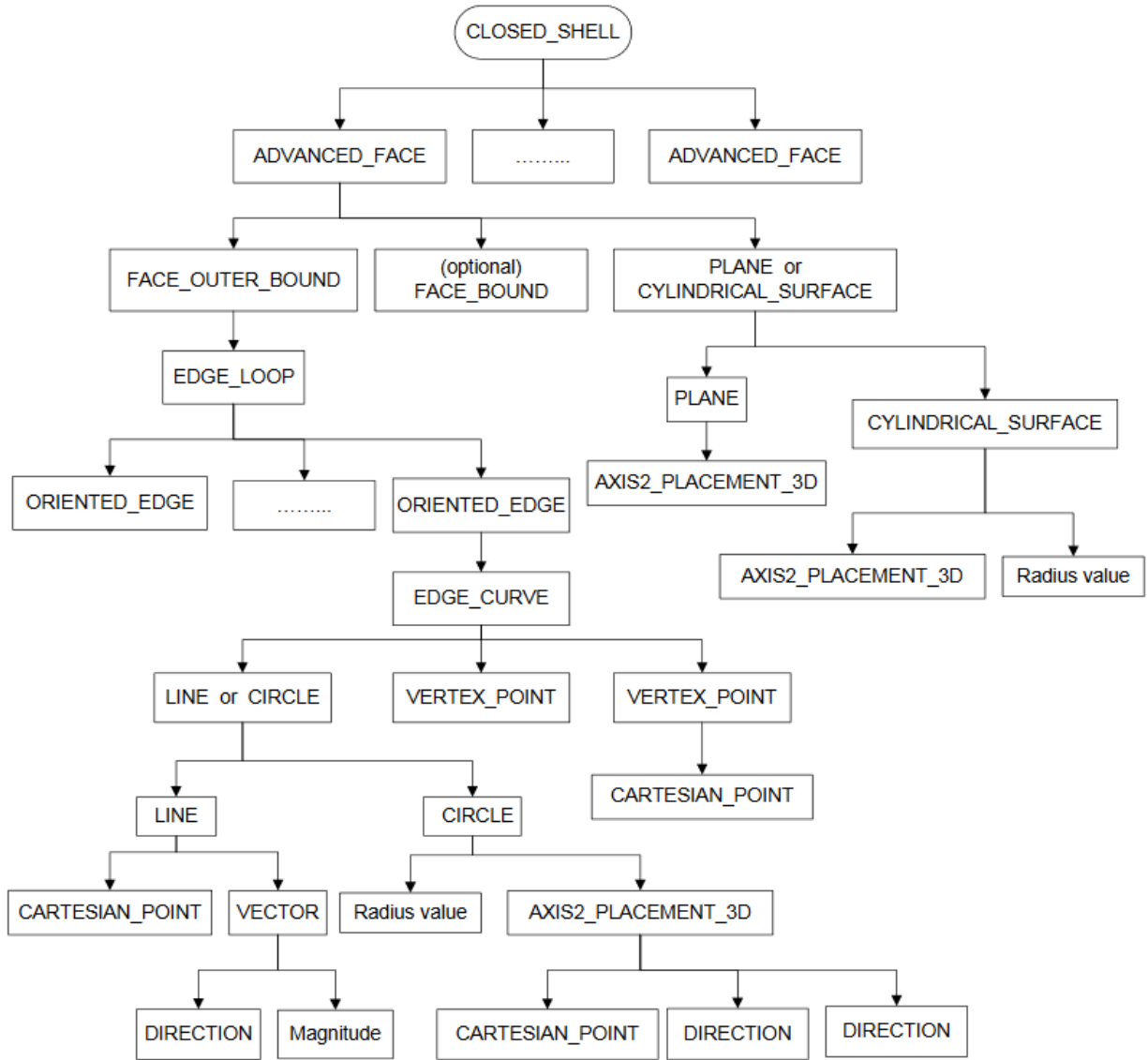


Figure 3: STEP AP203 Data Structure

[Source](#)

All entities listed on page 93 / Appendix 1 (Note: some entities listed are unrelated to AP 203) - [Recommended Practices for AP 203](#)

The CLOSED_SHELL entity represents an object, a project may have multiple closed-shells, each consisting of advanced-faces. These advanced-faces have their own sub-entities. The lowest most of these being Direction and Cartesian-point which contain geometrical information (X Y Z). The STEP file in the figures above displays a cube. Cubes have six sides and the file contains six

advanced-faces. A feature or face is derived from a hierarchical tree structure of entities linked forwards and backwards in the STEP file.

Example:

Step #18 in Figure 2 is as follows:

```
#18 = ADVANCED_FACE ( 'NONE', ( #108 ), #181, .F. );  
    #108 = FACE_OUTER_BOUND ( 'NONE', #99, .T. );  
        #99 = EDGE_LOOP ( 'NONE', ( #137, #58, #160, #69 ) );  
    #181 = PLANE ( 'NONE', #140 );  
        #140 = AXIS2_PLACEMENT_3D ( 'NONE', #222, #207, #205 );
```

ETC...

As you can see, #18 links to #108 and it links backwards to #99. Each step may or may not link to another step. This creates a tree-like structure with ADVANCED_FACE being the highest level entity. This tree-like structure is why XML could be useful in parsing the STEP file and extracting the geometrical data. (Chunxia Pan and Shane Smith, 2003)

XML

Extensible Markup Language (XML) is used to describe data. It is similar to HTML but instead of the pre-defined tags in HTML, XML allows you to define your own tags. This is very flexible and customizable. XML is said to be self-defined, this is because the structure is embedded with the data. There is no need to define any structure before retrieving the data.

XML uses elements defined by tags. These elements have a start tag and an end tag. All elements are defined inside a root element which is the highest level element. This concept is known as a nested element. You can take that many steps further and have elements nested inside those elements. This is what creates the tree-like structure similar to what we can see in the STEP file data structure above. The tags mentioned before can be used to describe the content of the data between the start and end tag. The value of XML can be seen when you have a large amount of information. This information if formatted correctly can be parsed and displayed in a more accessible way. (Margaret Rouse, 2014)

Euler Bernoulli Beam Theory

A beam is a structure in which one dimension is much larger than the other two. Euler's beam theory is a model of how beams behave under axial forces and bending. Dating back to 1750, Euler's beam theory has become a popular method as it relies on assumptions that give an answer that is close enough for design purposes and the model itself is less complex than others.

These two assumptions are;

- Plane sections remain plane
- Deformed beam angles are small

Euler's theory provides a means of calculating the load-carrying and deflection properties of beams.

Two Types of Load Distribution

(Note: Beam is fixed in one position only as CNC machines hold pieces at a single position)

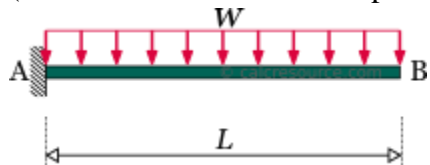


Figure 4: Uniformly Distributed Load (UDL)

[Source](#)

In this solution the load is evenly distributed across the entire length of the piece. It does not matter how many times W is divided.

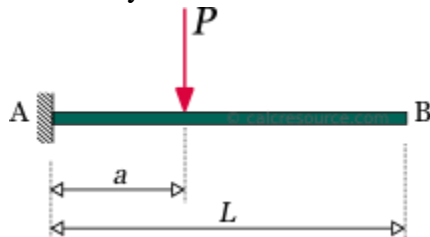


Figure 5: Single point load

[Source](#)

In this solution the load is placed on a single point only. The distance given.

The solution described in Figure 4 may be a more simplistic solution for the project to calculate the deflection and load carrying properties of the beam.

The required input is;

L - Length

E - Young's Modulus

I - Moment of Inertia

Type of loading - Uniform Distributed Load / Point Load

Results we need from this;

- Bending Moment
- Deflection

Vibration Analysis

Vibration Analysis is the process of monitoring the levels and patterns of vibration signals within an object. The goal is to detect abnormal vibration events, this helps determine the weak points of an object and the condition of the object.

Existing Technologies

Euler Bernoulli Simulators;

<https://www.fxsolver.com/browse/formulas/Cantilever+Euler+Beam+-+Displacement>

Euler function for beams with a single support and a load on the end point.

<https://calcresource.com/statics-cantilever-beam.html>

Euler function for fixed beams with a single support.

<https://calcresource.com/statics-simple-beam.html#anchor-2>

Euler function for fixed beams with two supports.

Technologies

Version Control Software

GitHub - I will be using github for this project, I chose GitHub as I have experience using it and I find it easy to use.

Documentation Software

Doxygen

C++

Familiarity - Basic

C is a compiled language developed in 1978. C++ is an extension of C and is a high level object oriented language created in 1998 by Bjarne Stroustrup. C++ differs from C, as C is a process oriented language. C++ can be used for a wide range of applications such as Video Games, GUI Apps, Web Browsers, Advanced Computations and Graphics and backend applications. C++ is applicable to all of these as it is a very fast language and provides a lot of control over hardware. C++ also tackles real world problems to model / calculate abstraction, encapsulation, interface and polymorphism. One of the main features of this language is the pointer. This allows you to allocate memory to be used and then reclaim that memory, but this process is not automatic. You have to clear the memory manually. If the memory is not reclaimed this may result in a run time error.

Libraries

Qt - <https://www.qt.io/developers>

QT has libraries that can be used to parse XML and create the GUI for this application.

ST-Developer - <http://www.steptools.com/docs/relnotes/relnotes-v15.html#overview>

ST- Developer has been used since 1991 to develop STEP and EXPRESS applications, it provides an EXPRESS compiler. This library also allows you to translate EXPRESS to XML or HTML. ST-Developer can also read files from any Application Protocol and can switch between them when writing. (This is not free)

C#

Familiarity - Basic

C# was created in 2000 by Microsoft to compete with Java. Similar to C++, C# is an object oriented language but it is a higher level language. It takes longer to compile as it is not as efficient. This is because C# generally has more libraries to be included. C# tends to be used more when you don't need the hardware to have high performance. It is modern, easy to use and the best choice for a simple web or desktop app. Unlike C++, C# allows you to automatically allocate and reclaim the memory. It is considered a 'safe' language as you get compiler warnings and errors that won't

allow you to damage your pc, while C++ will allow you to do anything as long as the syntax is correct.

Libraries

LINQ to XML - <https://docs.microsoft.com/en-us/dotnet/standard/linq/linq-xml-overview>

LINQ to XML provides more options and greater control over XML files when parsing / retrieving data.

Mono - <https://www.mono-project.com/docs/tools+libraries/libraries/>

Mono has libraries that provide GUI creation and XML parsing.

Conclusion

If this application requires the manual calculation of the Euler formula then C++ will be the optimal choice. But if the Euler formula can be outsourced, C# will be a good choice as the application will be simple and not tasking on hardware. It will be much easier to develop this application with C# as C++ would provide an unnecessary amount of control over the hardware. It would be better to not add that level of complexity.

API

Application Programming Interfaces (API) are software middlemen. They allow two applications to communicate with each other. For example when you use the weather app on your phone, the app is using an API to retrieve that information. An API may be required to simulate beam theory. Solid works offers such simulations as part of the application. If an API can be successfully used this would make it significantly easier to perform Euler's beam theory as you could simply extract faces from the STEP file and one by one subtract the features from a solid block and calculate results each time. Eventually finding the best order in which to cut these faces.

Solid works [API](#)

(Note: Solid Works offers support for C#)

The solid works simulation API has many different interfaces to pick and choose from. Only a couple are relevant to this project.

ICWBearingLoad Interface

Public Properties









	Name	Description
	BearingLoadUnit	Gets or sets the unit system for this bearing load.
	Direction	Gets or sets the direction of this bearing load in the selected coordinate system.
	DistributionType	Gets or sets the distribution type of this bearing load.
	UseTimeCurve	Gets or sets whether to use a time curve for this bearing load.
	XDirectionReverse	Gets or sets whether to reverse the X direction of this bearing load.
	XDirectionValue	Gets or sets the value of the bearing load in the X direction of the selected coordinate system.
	YDirectionReverse	Gets or sets whether to reverse the Y direction of this bearing load.
	YDirectionValue	Gets or sets the value of the bearing load in the Y direction of the selected coordinate system.

Figure 6: ICWBearing Load Interface (Public Properties)

[Source](#)

Public Methods











	Name	Description
	BearingLoadBeginEdit	Starts editing a bearing load.
	BearingLoadEndEdit	Ends editing a bearing load.
	GetEntityCount	Gets the number of entities for the bearing load.
	GetTimeCurve	Gets the time curve data for the bearing load of this dynamic study.
	InsertEntity	Inserts a source entity for the bearing load.
	RemoveEntity	Removes a source entity at the specified index from the bearing load.
	ReplaceCoordinateSystem	Replace the current coordinate system with the specified coordinate system.
	SetTimeCurve	Defines a time curve for the bearing load of this dynamic study.

Figure 7: ICWBearing Load Interface (Public Methods)

ICWDistributedMass Interface

Public Properties

	Name	Description
	TotalMass	Gets or sets the total distributed mass.
	Units	Gets or sets the units of the distributed mass.

[Top](#)

Public Methods



	Name	Description
	InsertEntity	Adds a face or shell edge to the collection of entities on which to distribute the mass.
	RemoveEntity	Removes a face or shell edge from the collection of entities on which to distribute the mass.

Figure 7: ICWDistributedMass Interface Members

[Source](#)

The Solid Works API allows you to import or create an object and perform various simulations on an object. The figures above show the properties you can customize and the methods you can apply to your piece.

You can see the Bearing Load interface which performs the action in figure 5 and the Distributed Mass interface which is relevant to figure 4.

Conclusion

From this research document we have observed how a STEP file's data structure works and possible methods to extract the geometrical data located inside. From these observations it was clear that STEP is organized in a format similar to XML in its tree-like structure of hierarchy. An XML parser isn't compatible with STEP unfortunately. A STEP to XML translator could be implemented but it might be easier to create a parser to catalogue the information and group together the steps that link together in the hierarchical structure. We also explored the different options for out-sourcing the Euler Bernoulli Beam Theory calculations. Further examination into the solid-works API and the online simulators is required. It was concluded that C# would be the optimal choice for this application as the solid-works API has C# support and the application won't be too tasking on the hardware.

Plagiarism Declaration

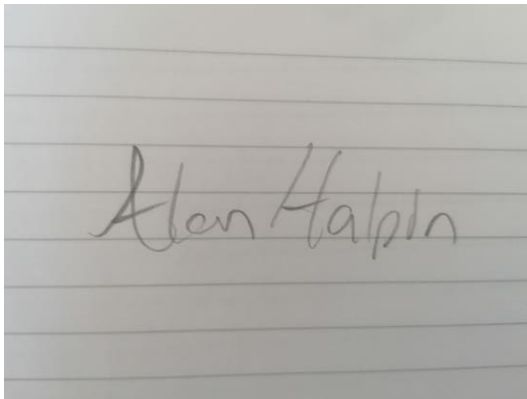
Declaration

- I declare that all material in this submission e.g.thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

Student Name: Alan Halpin

Student Number: C00229361

Signature:

A photograph of a handwritten signature 'Alan Halpin' on a piece of lined paper. The signature is written in dark ink and is centered on the page.

Date: 30/04/2021

References

Wikipedia (2020) ISO 10303-21 [online] available at https://en.wikipedia.org/wiki/ISO_10303-21

[Accessed: 10/2020]

Astro Machine Works (2017) What Is CNC Machining? An Overview of the CNC Machining Process [online] available at <https://astromachineworks.com/what-is-cnc-machining/>

[Accessed: 10/2020]

Datakit (2020) What is STEP? [online] available at https://www.datakit.com/en/step_what_is_step.php

[Accessed : 10/2020]

Torchmate (2020) How a plasma cutter works [online] available at <https://torchmate.com/white-papers/How-a-plasma-cutter-works>

[Accessed: 10/20]

STEP Tools, Inc (2020) Questions about STEP Application Protocols

[online] available at https://www.steptools.com/stds/stepnc/faq/faq_04.html

[Accessed : 11/20]

Margaret Rouse (2014) XML (Extensible Markup Language) available at

<https://whatis.techtarget.com/definition/XML-Extensible-Markup-Language>

[Accessed: 11/20]

Chunxia Pan, Shane Smith (2003) Extracting geometrical data from CAD STEP files [online] available at

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1080.6259&rep=rep1&type=pdf>

[Accessed 11/20]