2013

An integrated computer-aided modular fixture design system for machining semi-circular parts

Uday H. Farhan
Edith Cowan University

An Integrated Computer-Aided Modular Fixture Design System for Machining Semi-Circular Parts

BY

Uday Hameed FARHAN

A thesis submitted to the Faculty of Computing, Health and Science,

School of Engineering, Edith Cowan University

In fulfilment of the requirements for the

Degree of Master of Engineering Science

April 2013

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

ABSTRACT

Productivity is one of the most important factors in manufacturing processes because of the high level of market competition. In this regard, modular fixtures (MFs) play an important role in practically improving productivity in flexible manufacturing systems (FMSs) due to this technology using highly productive computer numerical control (CNC) machines. MFs consist of devices called jigs and fixtures for accurately holding the workpiece during different machining operations. The design process is complex, and traditional methods of MF design were not sufficiently productive.

Computer-aided design (CAD) software has rapidly improved as a result of the development of computer technology, and has provided huge opportunities for modular fixture designers to use its 3D modelling capabilities to develop more automated systems. Computer-aided fixture design (CAFD) systems have become automated by the use of artificial intelligence (AI) technology. This study will investigate the further improvement of automated CAFD systems by using AI tools. In this research, an integrated CAFD is developed by considering four main requirements:

- a 3D model of the workpiece,
- an expert system,
- assembly automation of MFs,
- an efficient feature library.

The 3D model is an important factor that can provide the appropriate specification of the workpiece; SolidWorks is used the CAD environment for undertaking the 3D modelling in this study. The expert system is applied as a tool to make right decisions about the CAFD planning process, including locating and clamping methods and their related element selection. This helps achieve a feasible fixture design layout. SolidWorks API and Visual Basic programming language are employed for the automating and simulation of the assembly process of MFs. A feature library of modular fixture elements is constructed as a means to simplify the fixture design process.

DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material

I also grant permission for the Library at Edith Cowan University to make duplicate copies of my thesis as required.

ACKNOWLEDGEMENTS

It is with immense gratitude that I acknowledge the help of my supervisor, Dr. Majid Tolouei-Rad. I am extremely grateful for his support, patience, wisdom and knowledge, despite his other academic commitments. I appreciate his help in solving any problems that arose during my study, and his understanding of my being away of my country and my family.

I owe my deepest gratitude to my father, Hameed Farhan, for his encouragement and support in the completion of my studies. Also I am pleased to have such a great father to look after my family. Without him this effort would be worthless.

I am indebted to my wife, Rabeea, for her love, support and patience for the three years I have been away. Thank you for your sacrifice, for caring for our children, I totally appreciate it. Thank you for standing beside me throughout my studies, you have been my inspiration and motivation to improve my knowledge and move my career forward, and I dedicate this work to you. I also thank my wonderful children, Asraa, Zahra, Ruquia, Asmaa and Younis, for always making me smile and for their understanding all those years while I was away instead of spending time with them. I hope that one day they can understand why I spent these years overseas and appreciate the benefits of that. Sorry to you all for the time I have been away. I hope this will start a new future for us.

In particular, I owe my thanks to Simona O’Brien, my best friend, my colleague, my neighbour. I cannot find words to express my gratitude for your time, support and trust. Thank you for opening your home to me, it was wonderful to become acquainted with your family, Liam, Kenneth and Jacqueline. Our time studying together helped me to overcome the enormous pressures I was facing during my studies. I really appreciate the time shared on happy and hard days, helping each other to complete our work, and opening a gateway to meet many new people from different academic fields. I will always remember our maxim: be happy, we will make it. Thank you for standing beside me during the writing of this thesis. It was an honour to work and study with you.

Also, I would like to thank my brother, my sister, my sister in law, my brother in law, and my step mother for their help and support for my family during my studies and being away.

Finally, I would also like to thank the following people:

- Professor Daryoush Habibi, head of the School of Engineering, for his time and advice, as well as the School of Engineering staff for their help.

- Sheavun Johnstone, the special person that I ever met, I am extremely thankful for her time, support and her friendship

- Ahmad Al Harahsheh, who helped me to be successful during my study at PIBT, I am pleased to have a friend like him.

- PIBT staff and teachers who provided me with the necessary knowledge of English, which I greatly value and never forget.

- Dave Langdon, ECU Student Connect Officer, for his help with any problem I had during my studies.

List of publications

On the basis of the outcome of this research the following papers have been published/accepted for publication:

Published in refereed international journals:

1- Uday Hameed Farhan, Simona O'Brien and Majid Tolouei-Rad, "SolidWorks Secondary Development With visual Basic 6 for an Automated Modular Fixture Assembly Approach". *International Journal of Engineering (IJE)*, Volume 6, Issue 6, 2012, pp. 290-304.

2- Uday Hameed Farhan, Majid Tolouei-Rad, and Simona O'Brien. "An automated approach for assembling modular fixture using SolidWorks". *World Academy of Science, Engineering and Technology*, Volume 72, 2012, pp. 394-397.

Accepted for publication in refereed international journals:

3- Uday Hameed Farhan and Majid Tolouei-Rad. "An integrated approach to assembling and automating Modular Fixtures design", *Emirates Journal for Engineering Research (EJER)*. Manuscript Number: EJER 37-12-S12.

Published in the proceedings of refereed international conferences:

4- Uday Hameed Farhan and Majid Tolouei-Rad, "Design of modular fixtures using a 3D-modelling approach". 19th International Congress on Modelling and Simulation, Perth, Australia, 2011, pp. 405-411.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

In the memory of my mother…

Chapter One

1. Introduction

1.1 Problem Statement

During the development of CAD/CAM (computer aided design and computer aided manufacturing) activities for manufacturing processes many issues need to be addressed, including the fixture design process for efficient machining. The traditional fixture design process cannot adapt to the rapid development in manufacturing technologies and their equipment, especially CNC (computer numerical control) machines [1]. Product quality, shorter production time, lower cost and efficient delivery are the main goals of manufacturing processes. To achieve these goals, fixture design is considered a vital factor and has a direct impact on machining operations [2]. Computer aided fixture design (CAFD) system was introduced as a solution for complex fixture design processes [1]. The initial use of CAFD was to apply CAD software as a tool for designing and assembling fixture elements by employing a standard fixture library [3]. However, automated CAFD systems have become more significant in manufacturing processes since the rapid improvement in CAM activities in recent decades. Automated CAFD allows the user to define a feasible fixture configuration, including locating methods and clamping mechanisms and layout for a given workpiece. Artificial intelligence (AI) technologies together with expert systems have been used to automate this process [4]. Different approaches of using these systems have been reported. Some studies used a genetic algorithm (GA) for the optimization of clamping and machining cutting forces, while others concentrated on the selection of locating and clamping methods and fixture elements by applying fuzzy logic or case-based reasoning (CBR).

Geometric considerations and machining features are essential factors in the fixture design process. Most studies focused on the features of prismatic parts, and many used AutoCAD as the CAD environment for carrying out the fixture design process. This study considers semi-circular shaped parts for representing an automated fixture design system, and glass container moulds could be a specific example for this kind of parts. A glass container mould consists of two pieces made with a high degree of accuracy due to the complexity of many glass containers. As most glass moulds have approximately the same shape (cylindrical or semi-cylindrical), the volume is the crucial factor that differs between moulds [5].

1.2 Research Questions

- SolidWorks is a powerful 3D modelling CAD software tool. 3D modelling is a crucial factor for verifying the geometry of machining features of a workpiece and its fixture elements. How can the 3D modelling capabilities of SolidWorks be employed to achieve the verification of the workpiece and fixture elements?

- Assembly relationships between modular fixtures (MF) elements are very important in the fixture design process. SolidWorks has assembly tools that make it an efficient environment for the assembly process. How can the assembly tools in SolidWorks be exploited to develop the assembly relationships of MF elements?

- MF standard elements are used in the fixture design process. To involve these elements in the design process it is important to build a feature library of these elements. SolidWorks has tools to build libraries. Is it possible to utilise this feature of SolidWorks to create a feature library of MF standard elements?

- Designers' experience is a key factor for automating the fixture design process. Expert systems could be used as a tool for building the expertise knowledge and to make the appropriate decisions for different applications. In this case, how can expert systems be applied in the fixture design process? In the case of the using other programs such as Visual Basic (VB), how can they be used for this purpose and be integrated with SolidWorks?

1.3 Aims and Significance

The 3D modelling approach is a key factor in developing assembly relationships between fixture elements and workpieces in the fixture design process. This approach should consider all the geometric and machining features of the workpiece. Therefore, in this study SolidWorks software is implemented as a CAD environment due to its strong 3D modelling and assembly capabilities.

It is important that automated CAFD systems reduce the design and production time. Therefore the main objectives of this research are presented here. First, to develop an integrated system that helps engineers to select the right fixture elements, as well as feasible and cost- and time-effective MF layouts for semi-circular workpieces. The VisiRule expert system is used as a decision making tool for this purpose. Second, the assembly process of these elements is automated by using SolidWorks API functions. An approach using ActiveX project in Visual Basic language is developed and integrated with SolidWorks API in order to create new menus in the SolidWorks environment for performing the automation process. As a result, the user can select the proper elements and perform the automation and simulation and then test the system before starting on machining operations.

Developing the feature library is the third objective of this research. In terms of completing the developed MF system, a feature library of standard elements is created in Microsoft Access format. This study also investigates SolidWorks capabilities for creating new 3D models of specific designed parts. With the use of DriveWorksXpress tool, 3D model features can be captured and modified to develop new models, and as a result a new fixture layout is generated. An approach of cutting forces calculation is presented in this study with the consideration of clamping forces as a means to achieve fixture layout stability.

These objectives are combined aiming the feasible layout of MF by automating the selection and the assembly of fixture elements. Other design's features of MF such as clamping forces optimisation and cost prediction could be investigated in further developments of the system.

1.4 Organisation of Thesis

This thesis is divided into nine Chapters. The first Chapter provides the general introduction, and the literature review for this research is presented in the second Chapter. This Chapter discusses previous studies on automated CAFD systems and their different approaches and methods. The Chapter highlights the reasons for applying the methods and techniques used in this research. The fundamentals of workpiece holding are conveyed in Chapter Three. Also, the principles of locating and clamping and their classifications are illustrated in more detail. Moreover, this Chapter highlights why the specific locating and clamping methods considered in this research are selected. The Chapter explains the importance of SolidWorks API and VB in the automation of the MF design process.

The approach to the selection of fixture elements and the proper locating and clamping methods are presented in Chapter Four. The knowledge base for this process is developed in this Chapter. The automation approach for MF design is explained in detail in Chapter Five, including the knowledge base and assembly procedure. More importantly, the approach to developing the VB project and integrating the SolidWorks API is covered. Chapter Six highlights the importance of using the DriveWorksXpress tool within SolidWorks, as well as the calculation of cutting forces for the chosen workpiece and MF layout. Chapter Seven offers the research conclusion, and important appendix files are contained in Chapter Nine.

Chapter Two

2. Literature Review

In this Chapter, a descriptive approach is adopted for the various elements of automating a CAD design system. In the final Section (2.2) this information is integrated to provide a context for the research in the thesis. Several researchers have focused on fixture design information representation. Mervyn *et al*, highlighted four models for representing fixture design information, namely conceptual design fixture ability feedback, intermediate part model fixture ability feedback, fixture assembly configuration and workpiece loading instructions [6]. A document type definition (DTD) approach was used for these models (**Figure 2-1**). Furthermore, some researchers introduced a fixture elements structure, defined the fixture element combination and introduced information about assembly relationships. Peng *et al*, provided a functional unit definition for modular fixtures (**Figure 2-2**), and introduced a method for extracting and evaluating the assembly relationships for modular fixture assembly design, including a definition of the assembly relationship constraints [7].

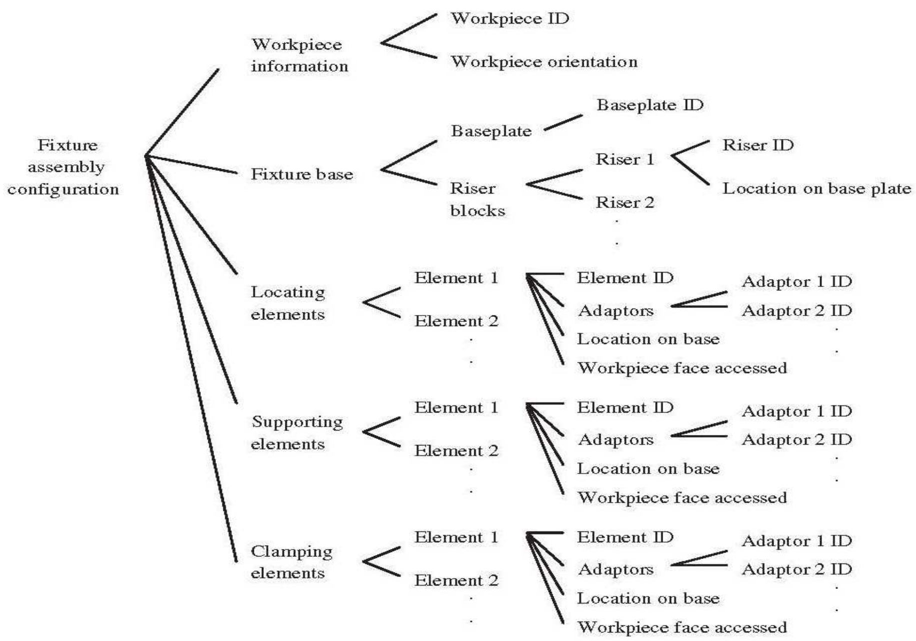


Figure 2-1. DTD file structure [6].

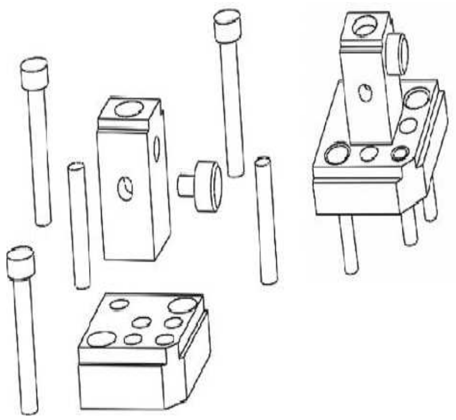


Figure 2-2. Typical functional unit [7].

2.1 Automated CAFD

As manufacturing processes rapidly developed, fixture design became an important factor in reducing the production time and costs, and CAFD systems were developed to simplify this design process. Automating CAFD systems has become crucial for competitiveness, as CAD/CAM activities improved and AI techniques were incorporated. However, and although many manufacturing activities are covered by CAM software such as tool paths generation, the full automation of CAFD systems still not achieved yet [8]. The challenge was how to computerise the human knowledge in order to automate these systems. In other words, put the tooling engineers experience in integrated software [8]. Many techniques and methods were used for this purpose including commercial tools as explained in the next sections.

2.1.1Genetic Algorithm

Genetic algorithm (GA) is a method that can generate solutions and optimise problems using natural evolution techniques. This method is based on a population of strings which encode candidate solutions in binary form. The population evolves toward better solutions. The process starts with individuals generated randomly in the population, and other generations then develop. The fitness of individuals in each generation is evaluated and a new population is formed. The process is terminated when a satisfactory fitness level is reached for the population, or a maximum number of generations is produced.

GA has been used as a specific tool to automate CAFD systems. Krishnakumar and Melkote used GA to optimise a machining fixture layout [9]. They used a 2D geometric approach for a given workpiece to perform the optimisation process (**Figure 2-3**). They also provided a method for implementing GA and explained how genetic operators work. In addition, they presented two examples of how the cutting process layout could be obtained.

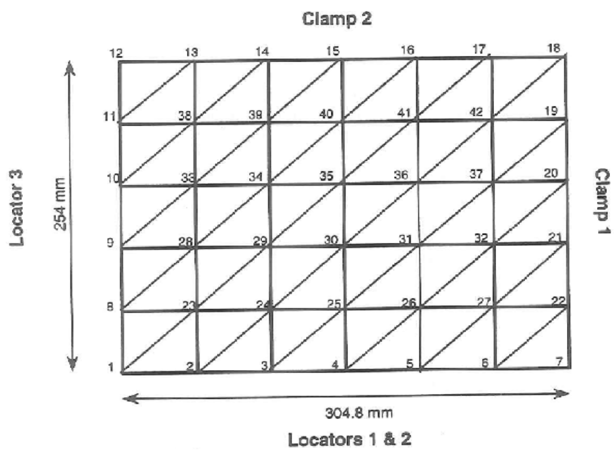


Figure 2-3. 2D geometry approach optimisation [9].

Chen *et al*, used GA by employing the direct search toolbox in MATLAB to optimise the clamping forces and control deformation during the fixture design [10]. For deformation calculation purposes, the ANSYS software package was used for specified cutting and clamping forces. In addition, they applied finite element analysis (FEA) to obtain the feasible fixture layout (**Figure 2-4**). Friction and chip removal effects were also involved, and a database was constructed to reduce computation time. This approach has been considered as a multi-objective method for controlling deformation.

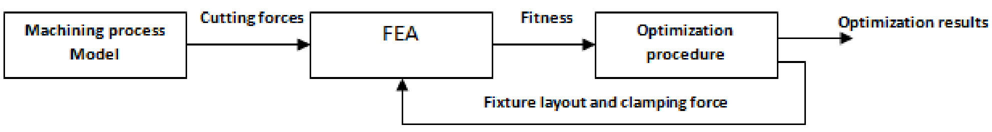


Figure 2-4. Flowchart of an optimisation process [10].

Hamedi offered an approach for combining GA and other tools including an Artificial Neural Network (ANN), and this method involved the use of FEA (**Figure 2-5**) [11]. The outputs from the FEA are passed to the ANN module. The GA module is then applied to optimise the set of clamping forces. These results are then integrated with computer-aided process planning (CAPP). Kaya provided a detailed method of how to calculate the deformation of the workpiece [12], in which first the GA representation and genetic operators were described, secondly GA was implemented as a basic approach with a combination of other programs, and thirdly ANSYS software and FEA were employed for 2D prismatic workpieces by applying a 3-2-1 locating method.

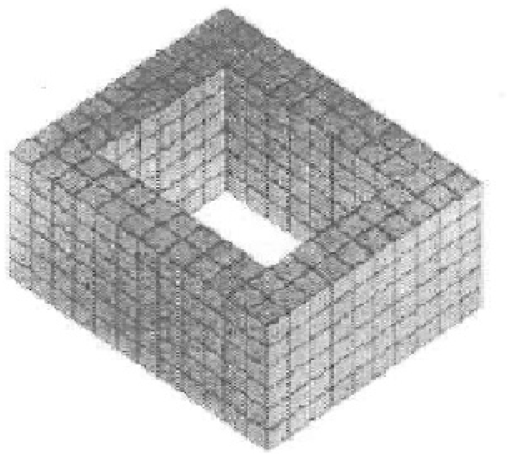


Figure 2-5. An FEA model [11].

A GA concept and FEA was also applied by Kulankura *et al*, for the optimisation of a fixture layout [13]. Cutting force calculations were implemented in their method.

2.1.2 Fuzzy Logic

Fuzzy logic is a method used for controlling a process by using the IF-THEN rule. This method consists of three stages:

1- Define fuzzy inferences (fuzzy sets and fuzzification);
2- write control laws (fuzzy inference rule base); and
3- convert the result into an engineering output (defuzzification).

Each value in the fuzzy set has a degree of membership within the set, varying from 100% (1) to 0% (0) and this differs from the crisp value (this can only be a true value while the others are false) [14]. In fuzzification, the input and output values are converted into their membership functions. A collection of rules that are related to the fuzzy sets is used to build the rule base, which is the controller of the system. A fuzzy inference engine is used to check the rules and find the corresponding outputs. Fuzzy inference is used to define a useful "engineering description" for each fuzzy descriptor. Several graphs are plotted from the fuzzification so that the membership degree of "different values in different fuzzy variables" can be described [14]. An example of fuzzy rules for the input variables, INPUT1 and INPUT2, and the output variable, OUTPUT, is:

IF INPUT1= Degree-of-membership in INPUT1-SET AND

INPUT2= Degree-of-membership in INPUT2-SET

THEN OUTPUT=Degree-of-membership in OUTPUT-SET

General forms of the base rules can be as follows:

If <condition>then <consequence>

If<condition1 and (or) condition2> then <consequence>

If <condition1 and (or) condition2>then <consequence1 and (or) consequence2>

Fuzzy reasoning was used by Martin and Lombard to introduce a method that defines a suitable positioning system for the workpiece [15]. They determined that the defect and length values of the datum surface and the part material are the main criteria for the positioning system. In this method, the concept of the IF-THEN rule was used to define the surface criteria for positioning. The process of this method was presented and a prototype was implemented using the "CAD X1 knowledge-based systems generator platform". Zhang and Peng represented an integrated system of fuzzy and rule-based reasoning (RBR) [16]. This system was divided into two stages, setup planner and fixture designer. The inputs for the setup planner were machining features, machining tool and operation selection. The output was the preliminary setup plan because the input was the fixture designer stage. The output of this stage is the final setup and fixture plan. This system is an integration of setup planning and fixture design in CAPP.

Guangfeng introduced an algorithm for workpiece locating design (**Figure 2-6**) [17]. They used the integration of RBR and fuzzy logic to achieve a feasible locating design. A fixture feature-based approach was applied based on a Pro/E 3D model that helped retrieve the fixture feature information and classify this information into locating and clamping categories.

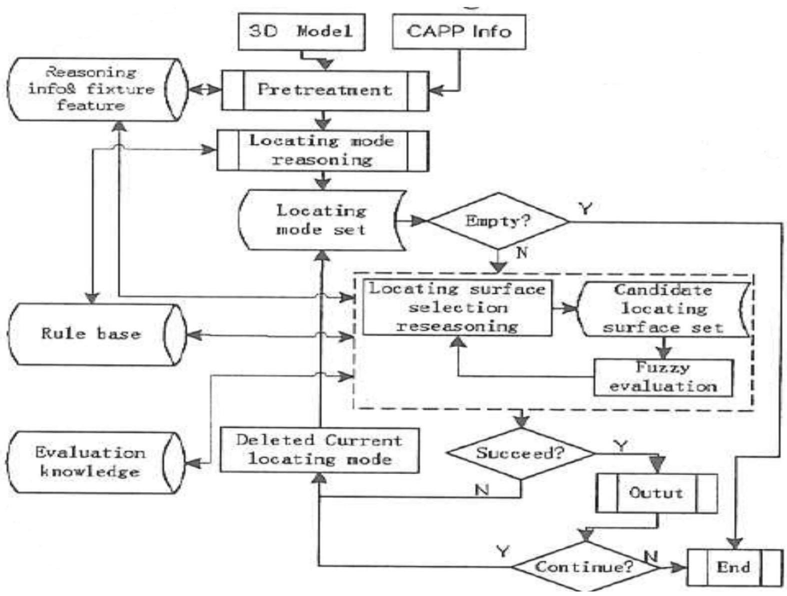


Figure 2-6. Flowchart of the feature based approach [17].

Kang *et al*, used a fuzzy evaluation method to choose locating schemes [18]. They employed fuzzy logic as part of their fixture design for networked manufacturing.

2.1.3 Case-Based Reasoning

Case-based reasoning (CBR) is a process that could be used to solve a problem based on solutions that are similar to related problems. It is a powerful computing reasoning method which is based on past personal experience. CBR consists of a four step process:

1- Retrieve cases from memory to solve a related target problem;
2- reuse a solution from the previous case;
3- test and revise the new solution; and
4- store the new case in memory.

CBR is viewed as a rapid problem-solving method that is adaptable to different applications [19]. One of these applications is the planning of a sequence of steps to achieve desired results. Moreover, CBR is regarded as a suitable method to find solutions for design problems. In general, design problems can be complex and may require input from an experienced designer. Diagnosis is another application for CBR, i.e. to provide explanations for given symptoms [19].

Different approaches have been employed for using CBR in fixture design. Kailing *et al*, applied a CBR reasoning machine in a CAFD system to choose similar cases [1]. In addition, they applied RBR and a knowledge base to obtain a feasible fixture design. This system consists of a fixture elements database, a knowledge base and a reasoning machine. The fixture elements database was classified into different categories, thus simplifying the fixture design process.

Boyle and Kevin presented a methodology called CAFixD (**Figure 2-7**) in which CBR was applied [2]. The objective of this methodology was to achieve the full fixture design including physical structure details of the fixture elements.

Another objective was to improve the CAFD process so that it could meet all the process requirements. The classification of fixture units and locating methods was also introduced.

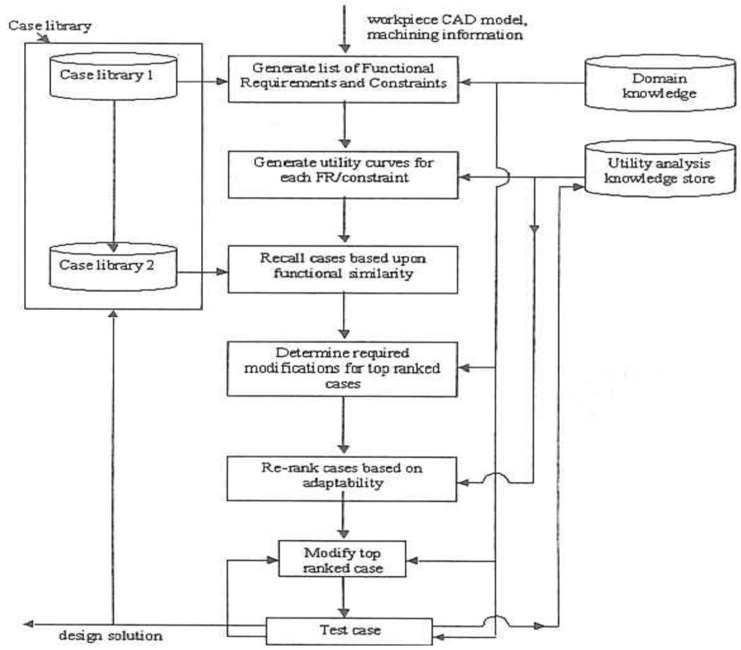


Figure 2-7. CAFixD flowchart [2].

CBR was implemented by Kang *et al*, [18] in their fixture design system for networked manufacturing. They also employed knowledge-based reasoning (KBR) as an assisting tool. They presented a hybrid model of CER/KBR for the fixture design process.

2.1.4 Expert Systems

Expert systems have been applied to automate modular fixture design processes. Kumar *et al*, used NEXPET–Object shell as a tool to represent the designer's knowledge when developing the rule-base for fixture design [20]. In their system, the locating, clamping, and supporting faces are determined for suitable fixture elements in order to automatically assemble them. Nee and Kumar also demonstrated a framework to automate the fixture design process by combining mathematical analysis with a rule-based approach [21]. The outcome of this system is the assembly sequence for the fixture elements. Nnaji and Alladin presented the structure of an expert system for computer-aided flexible fixture design [22]. They applied IF-THEN rules to develop the knowledge, and PROLOG as an expert approach to create the rules for the fixture system.

2.1.5 Other Methods

Some researchers have used different approaches to automate the fixture design process. Babu *et al*, implemented an automated modular fixture design system that used an AutoCAD platform [23]. They also used the AutoLISP program in combination with AutoCAD; this approach was based on 2D drawings. They introduced a methodology for the machining setup and the modular fixture automation sequence.

Kow *et al*, introduced the concept of semi-automated and automated fixture design [24]. Expertise knowledge was applied to define the rules and algorithms in a way that automated the process of determining fixture points and selecting fixture elements. The tool-collision-free fixture design was the main goal of this system, and three aspects of fixture design called interactive, semi-automated and automated were explained. In addition, this system showed how machining interference could be detected and a database of the parametric modular fixture elements could be used.

Ma *et al*, developed a fixture design system called FIX-DES, which was developed using C and C++ programs and a specific CAD environment [25]. The requirements of modular fixture configuration were implemented and the fixture elements were classified based on the feature concept. Assembly relationships were used in this system.

Kong *et al*, improved a CAFD system based on an AutoCAD software platform. VC++ and AutoLISP languages were employed in this system and a database of standard fixture elements was also presented (**Figure 2-8**) [26].

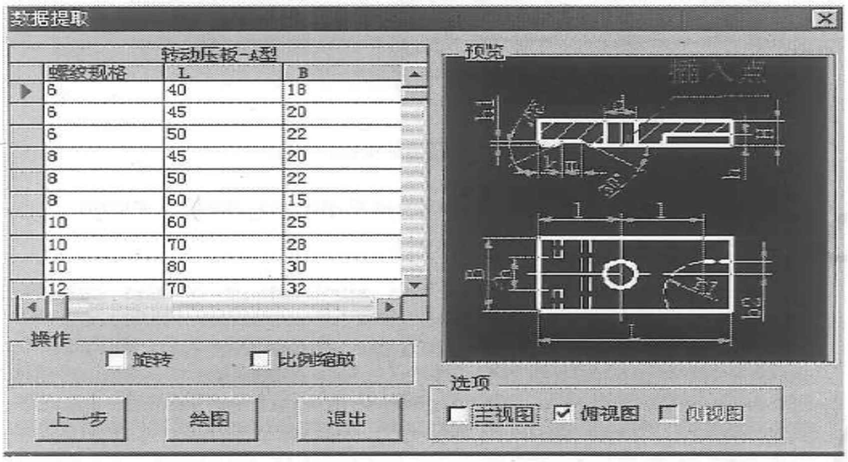


Figure 2-8. Standard fixture elements database [26].

Rong and Li developed a rapid fixture design system for an automated modular fixture configuration design [3]. This system enabled the interactive selection of locating and clamping methods and fixture components. The system was developed in a CAD environment by employing a pull-down menu of fixture designs, and different categories for locating and clamping methods and their sub-classifications were also presented. Assembly relationships among the fixture elements were employed in this system.

Shokri and Arezoo implemented a modular fixture configuration design for CMM (Coordinate Measuring Machines) measuring fixtures [27]. They presented a methodology and algorithm to develop the fixture units. This system was based on the SolidWorks platform, using VC++. SolidWorks was used for interference checking purposes between fixture elements and the workpiece. A procedure for fixture elements assembly was presented based on IF-THEN rules.

Hou and Trappey proposed a CAFD framework that could support different kinds of fixture design planning [28]. They implemented CAD software and a database management system (DBMS). A knowledge base was built as well as a fixture elements database that allowed more effective formulation of the data management process. This system integrated FoxPro and AutoCAD (**Figure 2-9**); furthermore AutoLISP with ADS toolkits were also used. Three levels of fixture planning were proposed; fixture data management, fixture elements selection and fixture layout planning. The concept of V-block planning for non-prismatic workpieces was also introduced, along with other fixture planning.

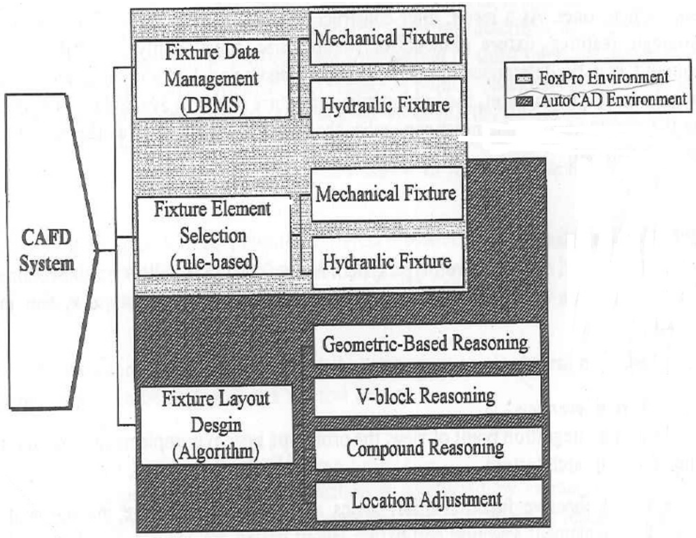


Figure 2-9. Integration of AutoCAD and FoxPro [28].

Dai *et al*, introduced a method for constructing a database of fixture elements and for building the fixturing towers [29]. This method used the knowledge-based system (ICAD) and UG-II (a graphical user interface) for workpiece modelling.

2.1.6 Commercial Tools

Fixture design is implemented in CAD/CAM software as well as CAPP systems. Lim *et al*, introduced a 3D modular fixture design expert (MOFDEX) system [30]. Their system included 3D solid workpiece definition, machining parameters, cutting tools, fixture elements CAD library, materials bill, and some other facilities. The authors created LISP/DBASE module for pricing purposes and LISP/CATIA module from a rule-based module for establishing the fixture components sequence and their coordinates.

Ajay and Tien dealt with the fixture planning in the automated process planning systems [31]. Their work included the use of Quick Turnorund Cell (QTC) system for the integration of process planning and fixture planning. QTC is a feature-based system and the authors used this tool to convert the design features to manufacturing features. The outcome of this system was establishing a fixture assembly providing the features for locating, clamping, supporting, and orientation.

Kenneth *et al*, presented an automated modular process planning system (AMPS) which implemented the setup and fixturing planning [32]. They planed the setup and the procedure of the fixture design and the relevant components. Also, they computed the location, clamping area, and other information for each fixturing configuration.

IMAO, a modular tool maker built a CAD/CAM plug-in which is called IMAO CAF system [8]. This system aimed to use 3D information from CAD software to the CAM software in an efficient way. By using this system, the user can specify the heights for clamping, supporting, and then selecting recommended elements.

2.2 Discussion of Literature Review

Fixture design is a complex process which is based on designers' experience. The traditional use of CAD software in CAED systems has brought significant limitations which can be addressed as follows:

- Manual selection of the fixture elements and the methods related to locating and clamping the workpiece. This takes a lot of time which is a critical factor in manufacturing processes.

- Use of 2D design of the workpiece and the fixture elements. This limits the verification of the machining features as well as the geometric features of the workpiece and its fixture elements.

- Lack of identification of the errors that could occur during the assembly phase of MF. This delays the achievement of the feasible fixturing layout of MF by considering the inappropriate assembly relationships between the workpiece and the fixture elements.

- Use of dis-integrated database of MF. It is important to implement a built-in database of the fixture elements to ensure the flexibility of CAFD systems.

From previous research as reported, GA has been applied to optimise the clamping and cutting forces for CAFD systems. GA is considered as a powerful tool for automating fixture design process; however it cannot automate the whole process on its own. To overcome this, other tools should be integrated with GA to automate the whole CAFD system such as ANN and FEA which were applied with GA for selecting fixture elements and optimising the deformation of the workpiece. Another issue is that GA is mostly used for 3-2-1 principle locating method which can only be applied to prismatic parts [12].

Fuzzy logic has been applied with various CAFD systems. This tool was implemented to automate the selection process for workpiece locating design and to define suitable positioning elements and locating schemes [16]. However, this method integrated with a RBR module in order to make the fixture design process more automated. Another limitation is that this method is more suitable for defining specific problems of MF layouts.

CBR seems to be an appropriate method for automating the process of defining the locating and clamping methods and related fixture elements. However, this method is limited by relying on similar cases to define the new design case. Therefore, similar to Fuzzy logic, CBR needs to be integrated with a RBR module to avoid this limitation by applying a knowledge-based approach for solving complex problems [1]. Another disadvantage is that most of studies that applied CBR in CAFD systems focus only on prismatic parts problem.

CAD software is another method used to automate CAFD systems. For instance, the AutoCAD platform is employed integrating with other programs such as C++ and AutoLISP to improve the CAFD system [26]. The main limitations of this method are the lack of 3D modelling capabilities of CAD software and the efficient integration with other programming languages in order to achieve the perfect assembly automation of MF.

Regarding the commercial tools, the major limitation is focusing on prismatic parts with the 3-2-1 locating method. Other limitations are the use of dis-integrated CAD library for the fixture elements and a library for specific components which are needed in the shop floor. For process planning, the systems are unable to achieve the optimum plans and the stages can be considered simultaneously [32].

Heading the aforementioned benefits and limitations of previous methods and in order to overcome existing limitations of present CAFD systems, the following techniques are employed in this study:

- VisiRule toolkit is applied to automate the selection process of fixturing methods and the appropriate fixture elements based on a knowledge base. This tool is used because it is an effective tool for building a decision making process based on flowcharts concept. In addition, this tool provides several rule formats which are useful for creating the knowledge base.

- SolidWorks software is used for designing and modelling the workpiece and the fixture elements. This software is used due to its powerful capabilities in 3D modelling and assembly. Moreover, SolidWorks has mating features that can be effectively employed to build the assembly relationships between the workpiece and the other components in CAFD systems.

- The integration between SolidWorks and VB programming language is generated to simplify the automation of the design and assembly tasks of MF, eliminating errors that could occur in the early design stages.

- A feature library of the fixture components is created in SolidWorks. This brings more flexibility for CAFD systems and significantly reduces the time needed for the design and assembly of MF.

Chapter Three

3. Workpiece Holding Fundamentals and Methods

The workpiece produced should meet its specifications, particularly in respect to shape, dimensions and tolerances, and to achieve this it is important to locate and clamp the workpiece correctly in the machine [33]. Some other factors also influence the workpiece holding configuration, including machining operation sequence, cost considerations, direction and strength of the cutting forces, and capabilities and orientation of the machine tools [33]. Moreover, fixture elements may be designed for a specific workpiece; and these are called dedicated fixtures. Conversely fixtures can be selected and combined from a database of standard fixture elements; and these are called MF [33]. There is a specific function for each fixture element, and a completed fixture structure can be built from a number of elements, including consideration of the types, classes and the functions that will lead to appropriate machining operations. This Chapter introduces the definitions of jigs and fixtures as basic workholding elements. Moreover, the types of these elements are explained as well as their functions. The concepts of locating and clamping were presented in detail along with the importance of modular fixturing and its classifications. The methods of fixture elements representation and structure were also presented. The fundamentals of the Visual Basic language and SolidWorks API were introduced in order to understand the functions of these techniques in developing an automated modular fixture design system.

3.1 Workpiece Holding Principles

3.1.1 Jigs and Fixtures

Jigs and fixtures are devices that can be used to hold, support and locate the workpiece accurately in order to maintain the required relationship between workpiece and cutting tool [34]. Hoffman defined a jig as a “special device that holds, supports or is placed on a part to be machined” [34]. The function of jigs is extended to the guiding of the cutting tool during the machining operation, and steel bushings are usually fitted with these devices (**Figure 3-1**). A fixture is defined as a "production tool that locates, holds and supports the work securely so the required machining operations can be performed". They are usually used with accessories, such as thickness gauges and set blocks to reference the cutting tool to the workpiece (**Figure 3-1**) [34]. Compared to jigs, fixtures "should be securely fastened to the machine stable which the work is done". Furthermore these devices can be used for different milling operations with the majority of standard machine tools [34].

According to Henriksen [35], jigs and fixtures provide many advantages for industry including:

1- Accuracy and interchangeability of the workpieces;
2- reduction in working time for different operations;
3- simplification of complicated operations; and
4- reduction in cost of assembly, maintenance and supply of spare parts.

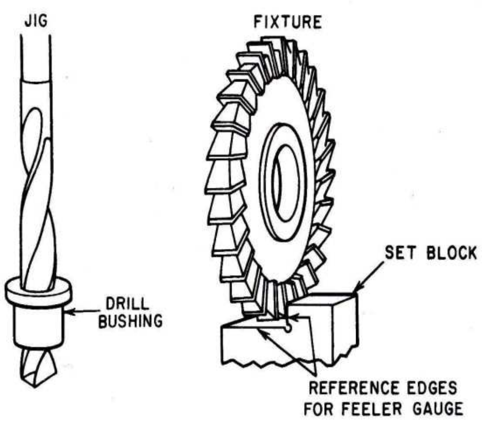


Figure 3-1. Jigs and fixtures referencing to the work [34].

As a result, the quality of the parts produced can be improved by using jigs and fixtures [35]. These benefits made the use of jigs and fixtures the solution to many problems in several industries. For example, in the aircraft and missile industries, geometry and dimensional problems have accelerated the use of jigs and fixtures. Moreover, jigs and fixtures are used for several machining operations, such as boring, broaching, drilling, milling, turning, tapping and facing; their advantages are exploited in many applications such as assembly, bending, heat treatment, inspection and testing [35].

According to Hoffman, jigs are classified into two general classes, boring and drill [34]. Boring jigs are used to make holes that can be considered “too large to drill or must be made an odd size”. Alternatively, drill jigs are used to “drill, ream, tap, chamfer, counterbore, countersink, reverse spotface or reverse countersink” (**Figure 3-2**) [34]. In addition, there are two types of jigs, open and closed. The difference between these two types is related to the number of sides of the workpiece to be machined. For the machining of only one side of the workpiece, open jigs are used; however, for more than one machined side, closed jigs are used [34].

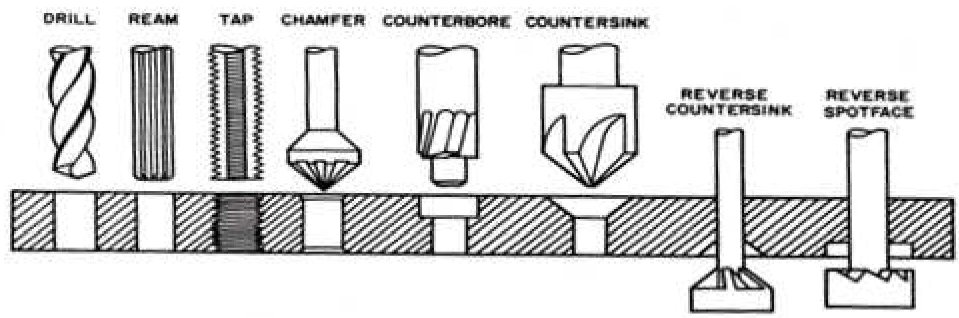


Figure 3-2. Common operations for drill jigs [34].

Hoffman also described some other types of jigs that were classified relating to the two main types, including the function of the jigs [34]. Examples of these types are template jigs, plate jigs, sandwich jigs, leaf jigs and pump jigs. **Figure 3-3** shows one example of these types. The same procedure for jigs is followed for the fixtures to be made; however fixtures should be stronger and heavier than jigs in order to resist larger cutting tools forces [34]. Hoffman introduced some types of fixture in relation to their functions; these are profiling fixtures, indexing fixtures, plate fixtures, angle-plate fixtures and multistation fixtures [34]. **Figure 3-4** shows one example of these types.

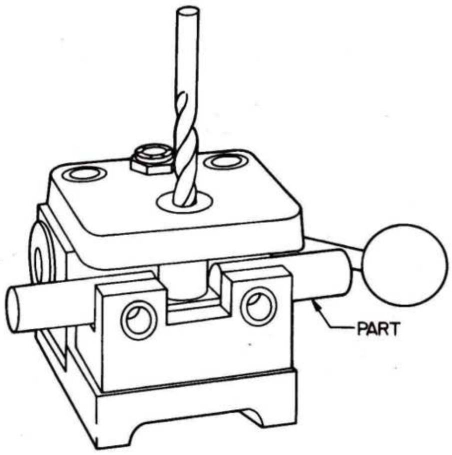


Figure 3-3. Pump jig [34].

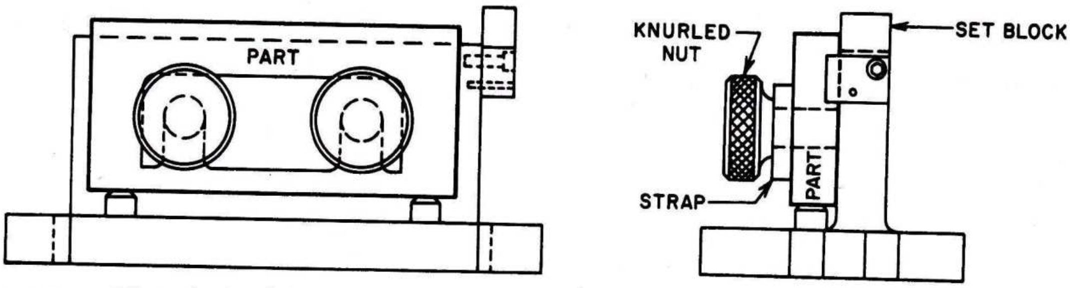


Figure 3-4. Angle plate fixture [34].

The main classification of fixtures is related to the type of machining process that uses the fixture. Their sub-classifications can be also defined. For example, if the fixtures are to be used on a milling machine they are called milling fixtures; and if the same fixtures are used to perform straddle milling, they are called straddle-milling fixtures; the same principles are applied to the lathe machine [34].

3.1.2 Locating Principles

Locating can be defined as a process for positioning and orientating the workpiece on the machine in order to complete the desired machining process [36]. For specifying the most appropriate and effective locating for the workpiece, it is important to understand the degrees of freedom (DOF). In space there are twelve DOF (**Figure 3-5**). These DOF are related to the central axes of the workpiece with “six axial DOF and six radial DOF” [37]. Axial DOF refers to the liner movement of the workpiece, while radial DOF refers to the rotational movement of the workpiece.

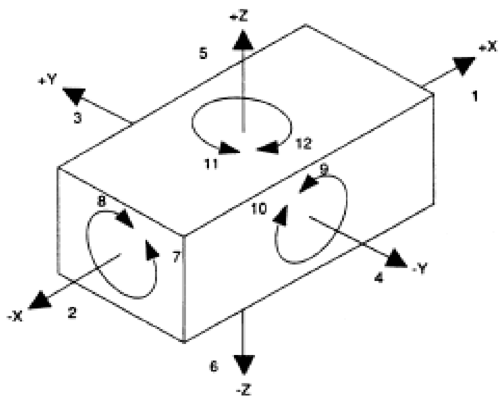


Figure 3-5. Twelve degrees of freedom in space [38].

To restrict the movement of the workpiece, locators are used, and they should be strong enough against the cutting forces to maintain workpiece’s position [37]. In general, three location forms can be considered, namely plane, concentric and radial [38]. For plane locating, locators are used to locate the workpiece on any surface such as flat, circular or irregular; while for concentric locating, pin-hole locators are used (**Figure 3-6**). For radial forms, locators are utilised to restrict the workpiece’s movement around the concentric locators [38]. In many modular fixture systems, a combination of these three forms can be used for performing the locating function. After determining the locating form for the workpiece, the designer should ensure that this locating can be applied to similar workpieces by selecting the appropriate locators which meet the needs of producing interchangeable parts [36]. The common locating method is 3-2-1, which considers three perpendicular surfaces of the workpiece to define the locating position [36]. The locating position is accomplished by defining three locating points on the first surface and two locating points on the second surface, which is normally perpendicular to the first surface; finally one locating point is defined on the third surface, which is perpendicular to the other two surfaces [36]. **Figure 3-7** shows the 3-2-1 locating method.

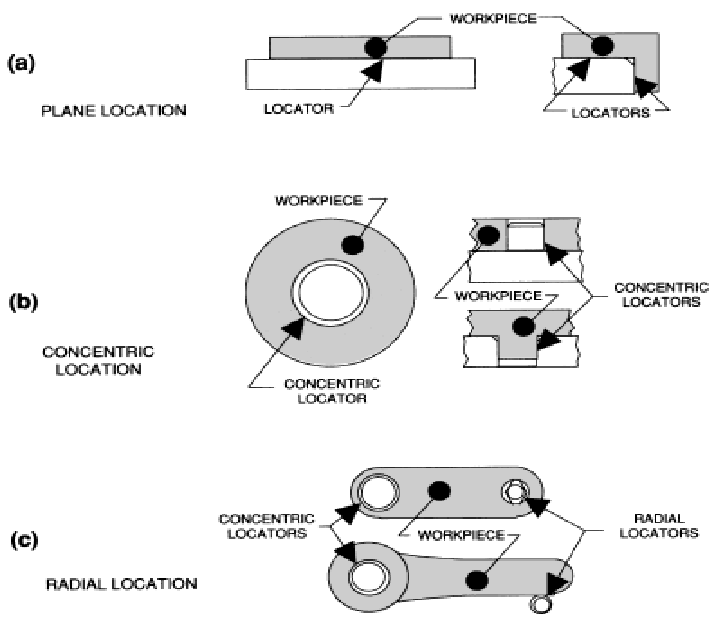


Figure 3-6. Three locating forms [38].

Alternatively, the internal surfaces of the workpiece can be considered for locating purposes. In this case, concentric and radial locating forms are applied with locating pins and locating plugs. The locating pins are used for smaller holes while locating plugs are used for larger holes [38]. **Figure 3-8** shows concentric and radial locating.

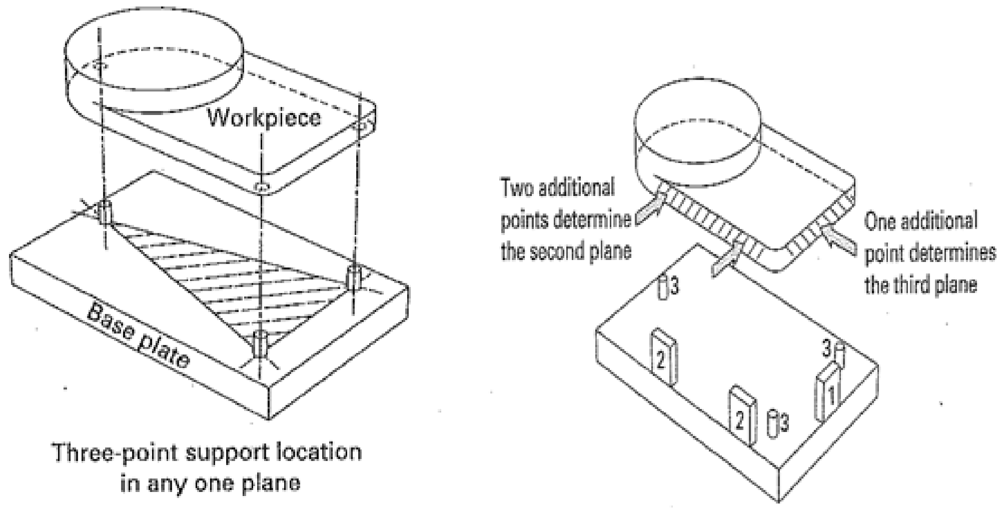


Figure 3-7. 3-2-1 principle [36].

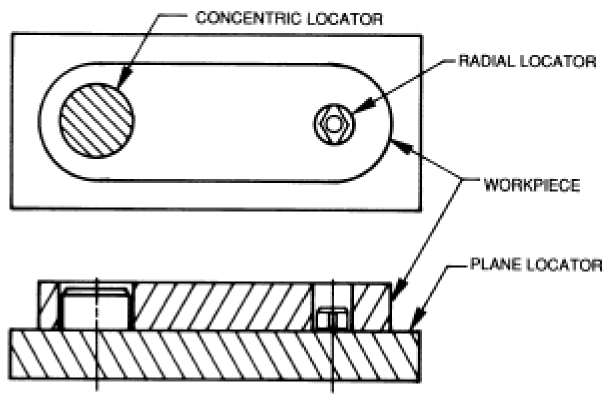


Figure 3-8. Using concentric and radial locators [38].

Rong and Li divided the locating methods into five main categories, namely 3-2-1, plane and pin-hole, long-pin, V-block and V-pad locating, with further variations for each of these categories [3]. For plane and pin-hole locating method, a plane surface of the workpiece is considered as a primary locating surface and an inner cylindrical surface is used as a secondary locating surface. For the long-pin locating method, an inner cylindrical surface of the workpiece is used as a primary locating surface to restrict four DOF. The V-block locating method is appropriate for primary external cylindrical surfaces as a primary locating surface, while V-pad is used for external cylindrical surfaces as a secondary locating surface [3].

3.1.3 Clamping Principles

Clamping can be defined as a mechanism for holding the workpiece against the locators and for resisting the effects that occur due to the cutting forces (**Figure 3-9**) [4]. The direction of clamps should be considered according to the directions of the cutting forces, and it is an important factor for securely holding the workpiece during machining operations [4]. However, clamps will not resist the "primary cutting forces"; they keep the workpiece position against the locators and resist the "secondary cutting forces" (**Figure 3-10**) [38], while locators will resist the primary cutting forces. Moreover, cutting forces can be predicted by the designer and this will help reduce the "magnitude of the required clamping stresses" [36]. The size of the clamp is another factor that should be considered in the fixture design; the clamp should be large enough to hold the workpiece securely and, in contrast, it should be small enough to stay away of the tool path to prevent any collisions from occurring [38].

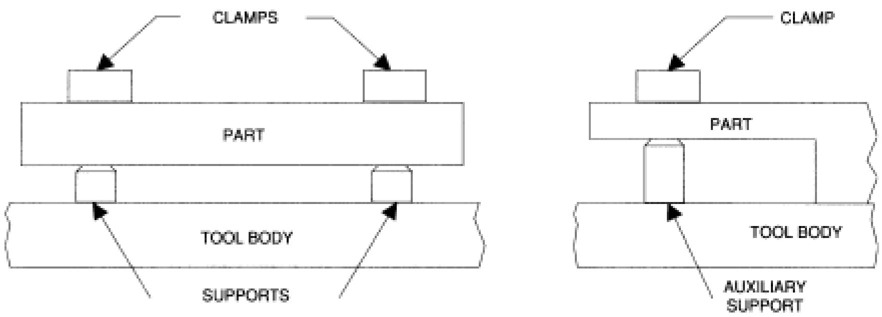


Figure 3-9. Clamps against locators [38].

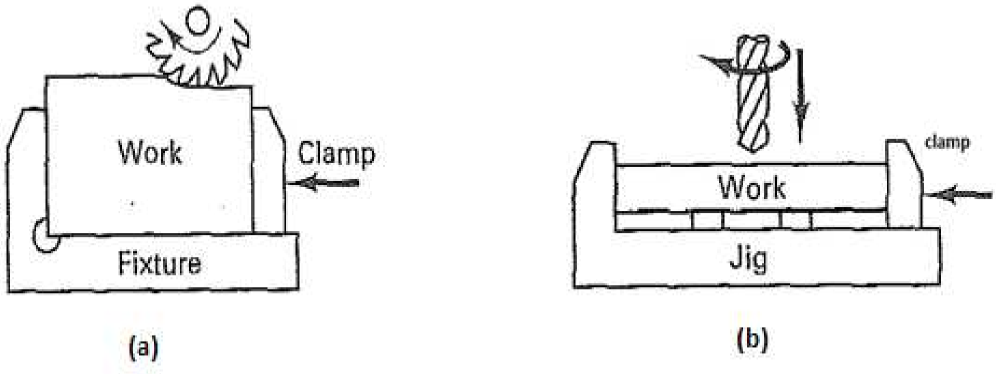


Figure 3-10. Two different cutting forces, a) milling and b) drilling [36].

Figure 3-11 shows the effect of clamp size by considering gooseneck clamps instead of a standard strap. In addition, other factors including holding the workpiece securely under vibration, loading and stress, damage prevention to the workpiece and improving load/unloading speed should be taken in consideration for clamping [38]. For modular fixture systems, two major categories were defined for clamping methods; these are top and side clamping (**Figure 3-12**) [3].

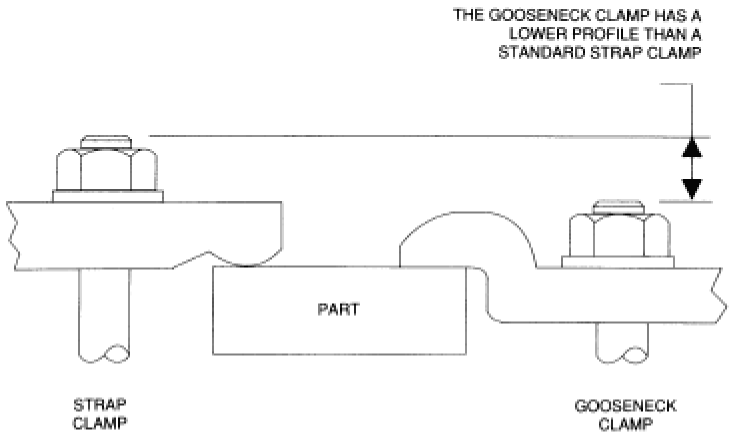


Figure 3-11. Effect of clamp height [38].

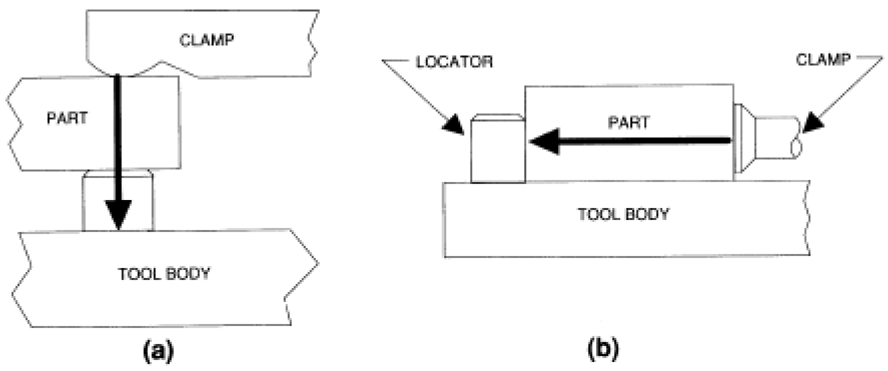


Figure 3-12. Two types of clamping, a) top clamping, b) side clamping [38].

3.2 Modular Fixturing

Modular fixturing can be defined as a system for building several combinations of standard components that can serve a wide variety of workpieces. These fixture elements can be assembled and reused in order to generate different constructions of jigs and fixtures [34]. The building process of modular fixturing system depends on selecting the necessary fixture components to be assembled, and this assembly process will be the basis for building “more detailed systems”. As a result, fixture elements can be built for any kind of workpiece by using modular fixturing systems [34]. The use of modular fixturing results in a reduction in the design and assembly times by eliminating the use of “dedicated fixtures” and their special components [33]. Moreover, the database of MFs contains the necessary design information with mating features for standard elements, and it is easy to modify [33]. Modular fixturing systems were classified into three major groups based on the construction type: Subplate, T-slot, and dowel pin [34].

3.2.1 Subplate Systems

These systems consist of a series of components that can be assembled together to build several workholding combinations. Furthermore, these components can be used separately for different purposes. **Figure 3-13** shows some of these components. Subplates are available for both T-slot and dowel pin collections. Therefore subplates can serve any kind of "fixturing operations" [34]. However, this kind of modular fixturing system is mainly used for workpieces of very large size [34].

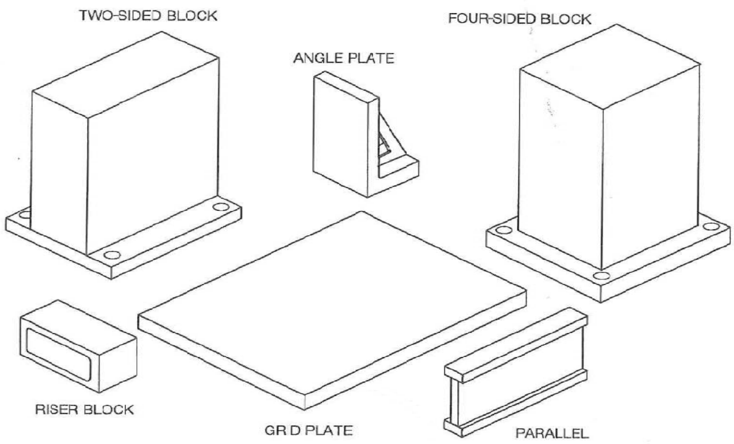


Figure 3-13. Subplate system components [34].

3.2.2 T-Slot Systems

These systems use fixture elements which are machined with T-slots (**Figure 3-14**). They are considered as strong and adaptable systems, and T-slot modular elements can be easily positioned. However, these systems need perfect measurements for the positioning of the fixture elements and for defining the reference points [34].

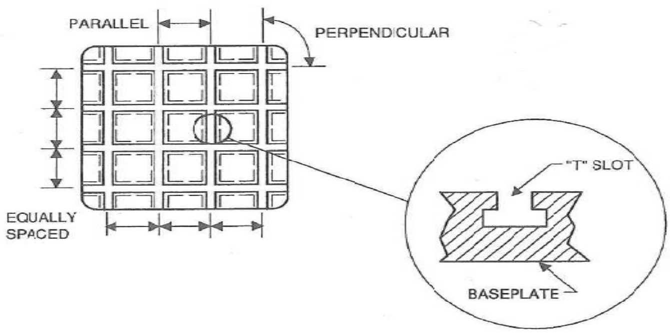


Figure 3-14. T-slot baseplate [34].

3.2.3 Dowel Pin Systems

These systems are made with holes in a grid pattern for positioning of the fixture components (**Figure 3-15**). Dowel pin systems have positioning capabilities that make the locating of the fixture elements easier and faster, because each hole is considered a locating point and, as a result, each hole is a fixed reference point [34]. One problem with these systems is in the adjustment of the clamping positions because the holes are fixed points. The solution for that is by making slots in the fixture elements in order to adjust their position in the workholding system [34].

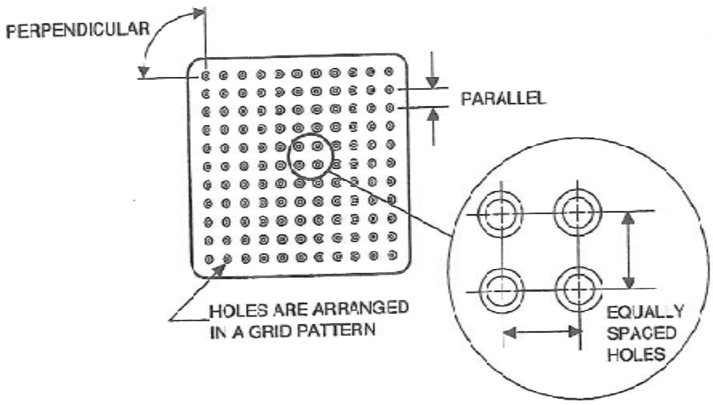


Figure 3-15. Dowel pin baseplate [34].

Dowel pin systems come with two different patterns of holes. The first pattern is made of both dowel pin holes and tapped holes. The second one is made of only dowel pin holes. For the first kind, the two kinds of hole serve for locating and clamping purposes of the fixture elements. While in the second one, the same holes are used for positioning and clamping mounting [34]. **Figure 3-16** shows the difference between these two types.

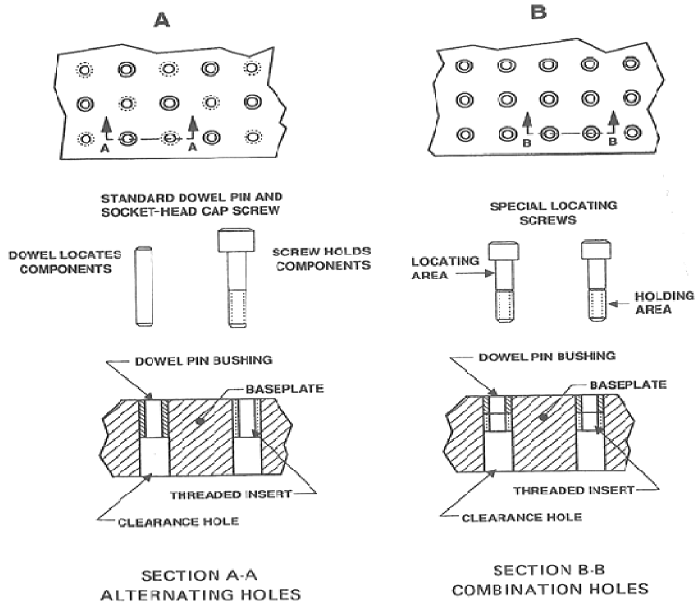


Figure 3-16. Differences between T-slot and dowel pin systems [34].

3.2.4 Modular Fixturing Building

Modular fixturing systems consist of a set of standard elements, such as baseplates, locators, supporters, clamps and all the other accessories (**Figure 3-17**). By assembling these elements together, suitable workholding systems for a variety of workpieces can be achieved [34]. The assembling process starts with selecting the baseplate depending on the size of the workpiece. Then the locating elements are chosen and assembled onto the baseplate. After that, the clamping elements are selected to suit the chosen workpiece. Finally, the other elements and accessories are added for completing the modular system [34]. After building the workholding modular system, the machined operations are started in order to produce the specific part or workpiece. When the part production process is finished, the modular system is disassembled and the elements are sent to the store to be used for the building of other modular workholding systems [34].

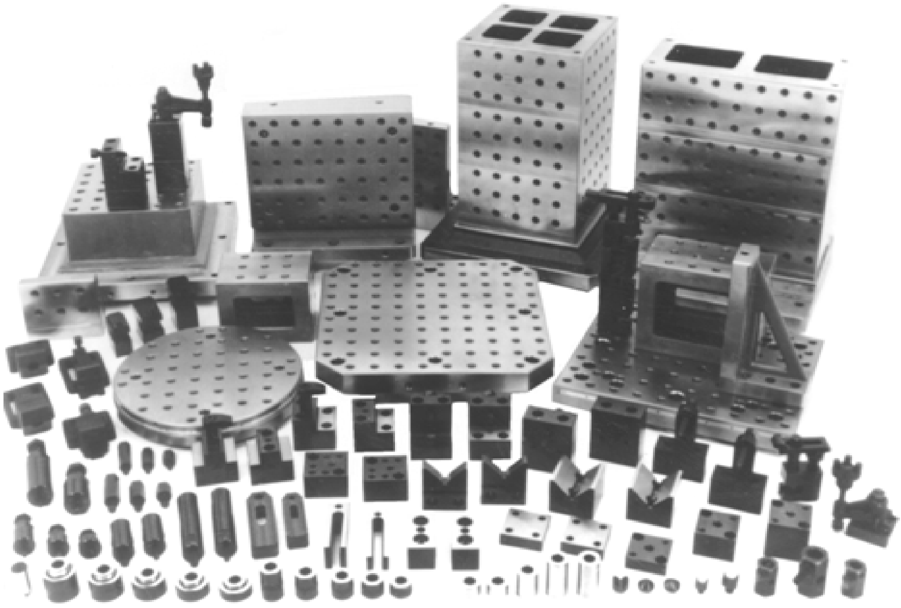


Figure 3-17. Modular fixturing standard elements [34].

Figure 3-18 shows an example of modular fixturing systems. The advantages of reusing the modular fixture elements lead to reducing time and costs. One hour of the modular fixturing building process equals about six hours of “conventional jig or fixture” building [34].

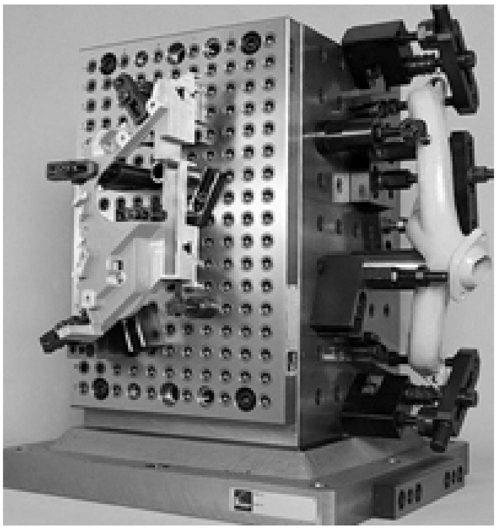


Figure 3-18. Modular workholding system [39].

3.3 Implementation Techniques

The main purpose of the automated modular fixture design is to provide a system that allows for:

1- Selecting fixture elements from a database in terms of assembly then according to the fixturing points.

2- Planning the fixture assembly process by determining a reasonable assembly sequences which allow assembling and disassembling of fixture components. The architecture of the Modular Fixture Assembly Design System (MFADS) is shown in **Figure 3-19**.

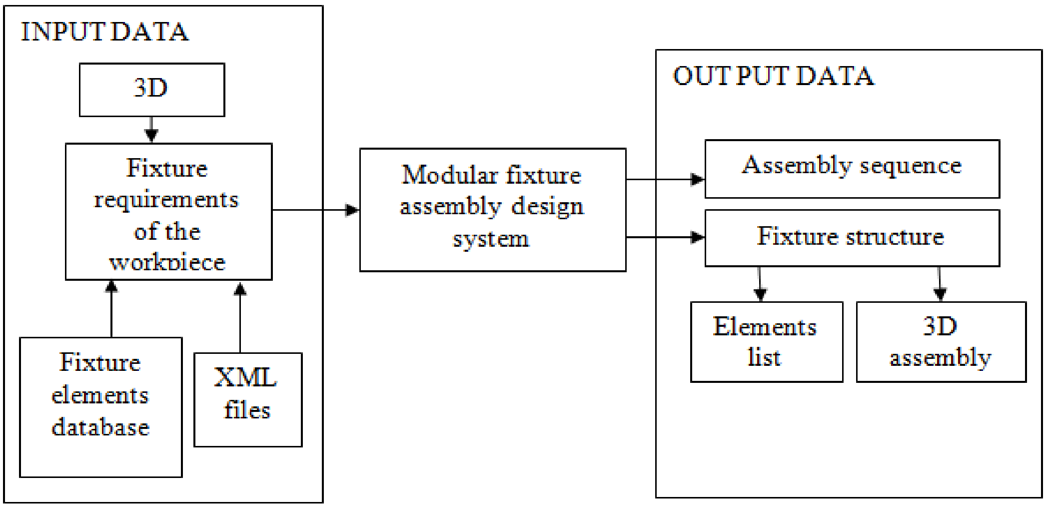


Figure 3-19. MFADS general architecture [40].

In order to explain the role of this system and how the designers can exploit it effectively, it is important to define the stages that must be followed for performing the task. There are five stages, namely element selection, structure design, element assembly, modification and adjustment, and assembly planning [40]. **Figure 3-20** shows these stages.

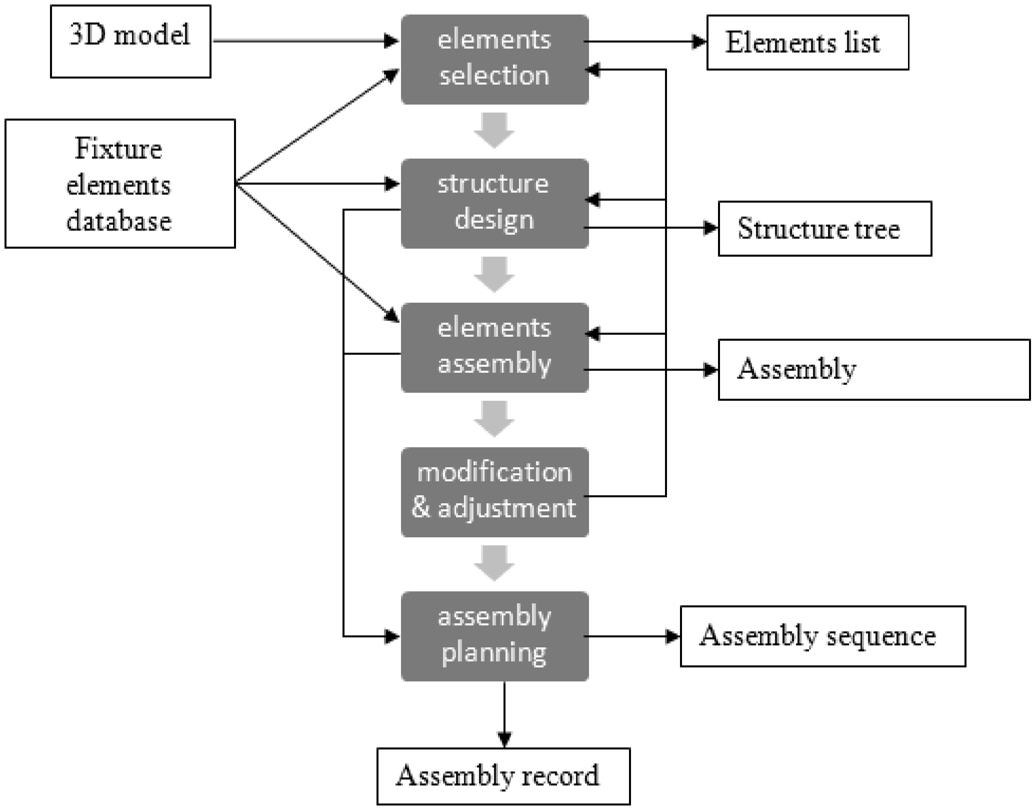


Figure 3-20. Function pattern for the main stages of MFADS [40].

As shown in **Figure 3-20**, a modular fixture elements database should provide information about element function, element dimensions and element assembly features for the first three stages, elements selection, structure design and elements assembly. The results of these three stages are elements list, assembly structure tree and assembly relationships. A modification and adjustment stage is responsible for providing feedback about element replacement, structure design modification and element assembly adjustment to ensure that the results from the first three stages are correct. After that, the designer will use the information on the structure design and element assembly stages for performing assembly planning and, as a result, the assembly sequence and assembly record can be completed.

In order to implement the proper fixture assembly design system there are four main factors that should be considered by the designer to effectively complete the whole assembly process. These factors are information representation of modular fixture, fixture element structure, assembly process methodology and the database of fixture elements.

3.3.1 Information Representation

One of the most important problems that must be considered by the designer is how to represent the assembly model [7]. In addition, fixture geometric and engineering information, as well as topological relationships and constraint information, need to be represented in the assembly model. In order to fulfil real time display and collision detection, the objects are often represented as a polygon model. However, this model loses the topological relationships as well as the engineering information, and without this information it is difficult to apply the assembly process [40]. Moreover, information representation is necessary in fixture assembly/disassembly processes. Therefore, developing an appropriate model representation is important to support modular fixture assembly design [7].

Fixture element representation includes the important information that is required in the assembly process [40]. In order to accomplish this, assembly feature relationship information must be considered by the designer.

In assembly feature relationship information, designers must represent the mating relationships among the different elements in the modular fixture system. In this research, a hole-dowel system is considered and it consists of a baseplate with holes and the elements are located by holes and dowels.

Because the fixture elements are highly standard parts, a fixture scheme can be used to describe the modular fixture elements [40]. After establishing this scheme, the designer selects the appropriate standard elements. Then, these elements should be assembled in a reasonable way for configuring the fixture system [7]. In general there are eight function faces that are determined as assembly features of fixture elements, namely supporting faces, supported faces, locating holes, counterbore holes, screw holes, fixing slots and screw bolts [25].

To illustrate an assembly feature position, a mark can be identified to complete the "related position of two assembled elements when an assembly operation is carried out" [7]. The definitions below can be followed:

- Mark: M_{ar} --- (P_{nt},V_{ec})
- Point: P_{nt} ---(x,y,z)
- Vector: V_{ec} ---(v_x,v_y, v_z)

Figure 3-21 shows the mark definition for plan and cylinder features. The designer should determine the mark for each assembly feature, and this mark should be stored as a requirement of the element model [7].

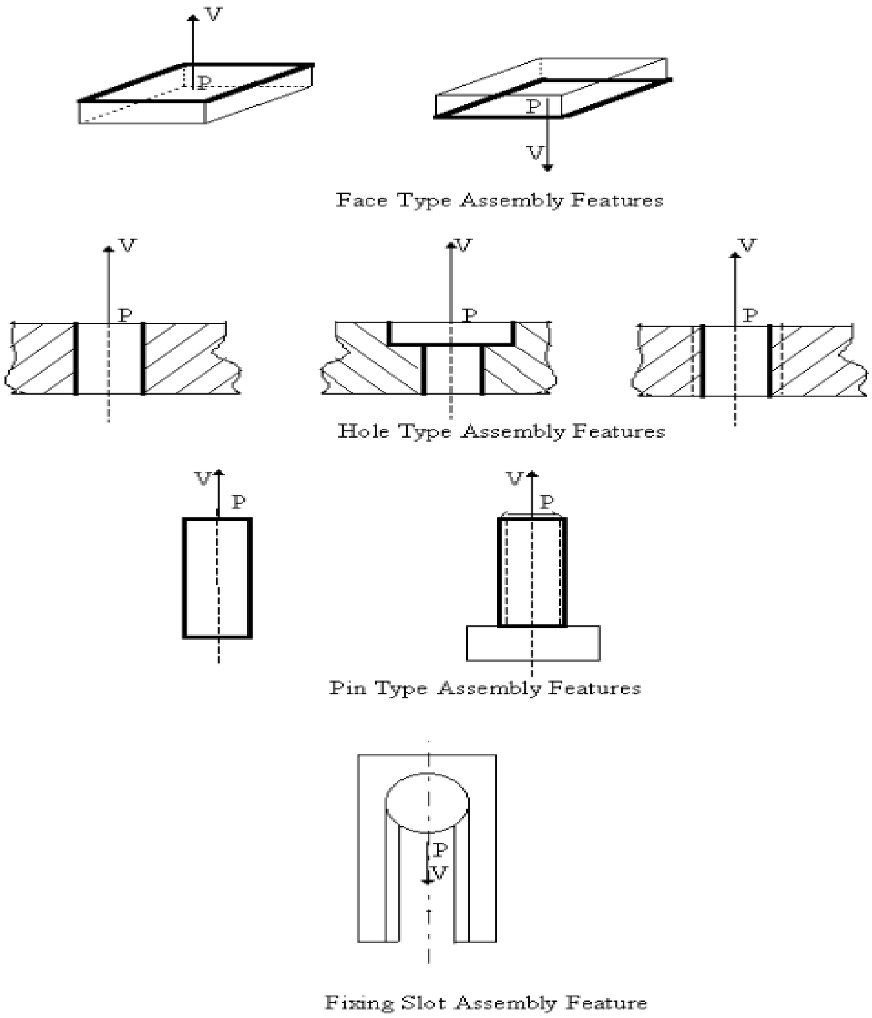


Figure 3-21. Different mark definitions for assembly features [41].

In general, mating relationships can be divided into five types, namely against, align, fit, screw fit and across (**Figure 3-22**) [42]. These mating relationships can be considered as a base for describing the "possible assembly relationship of any two assembled fixture elements" [40]. In addition, there are two basic geometric elements called line and plane which can be used for illustrating the assembly relationship of fixture elements, and the constraints of these two elements can be considered for describing the mating relationship for any two assembly features [40].

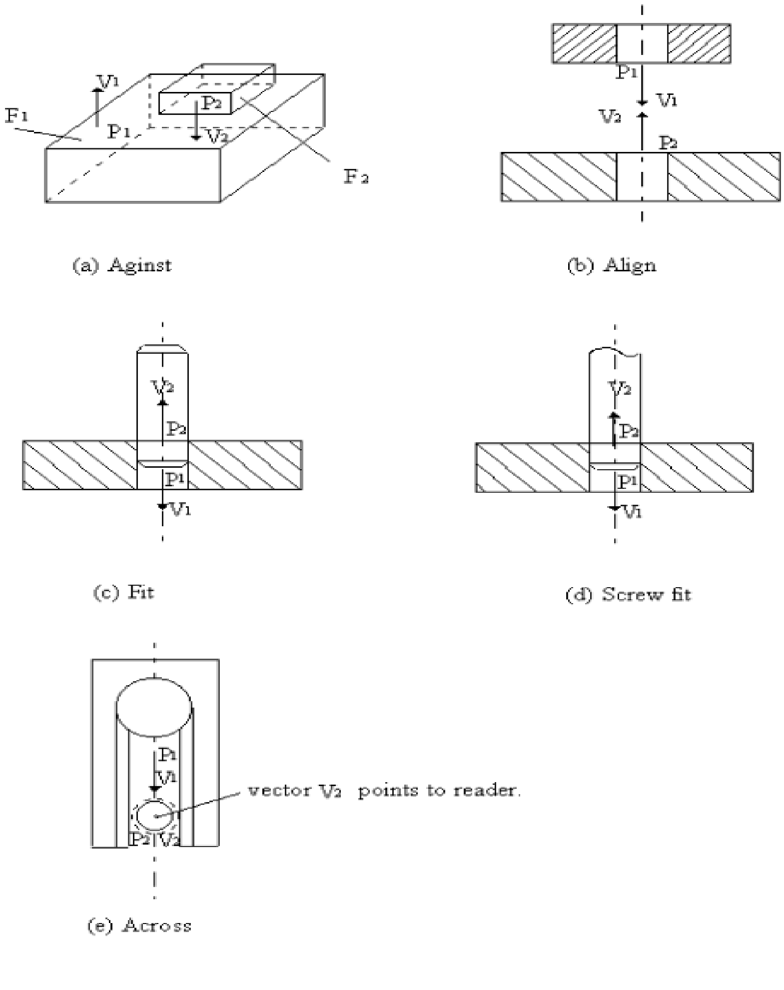


Figure 3-22. Five types of mating relationship [41].

3.3.2 Fixture Structure

After representing all the necessary information about fixture elements and assembly features, it is important to build a proposed structure which can illustrate the relationships between all the elements and how they are assembled. In a modular fixture assembly system, an individual fixture element cannot perform the role alone; it must be assembled to another element for defining the relationship and to define the aim of this relationship. For example, a clamp must be assembled to a baseplate in order to perform a clamping function. Moreover, fixture elements can be assembled in groups, and these groups are called “functional units”.

Each functional unit consists of one or more elements and these units are responsible for supporting the workpiece on the baseplate. The functional units can be divided into three types according to their function, namely locating, clamping, and supporting [43]. The structure of the fixture elements illustrates the relationships between the elements and how they are assembled. In MF design an individual fixture element is to be assembled to other elements, and the relationship between these elements in addition to the purpose of the relationship should be explicitly defined. **Figure 3-23** shows the structure of the fixture elements for connecting the workpiece to the baseplate. **Figure 3-24** illustrates a typical sample of the functional element unit.

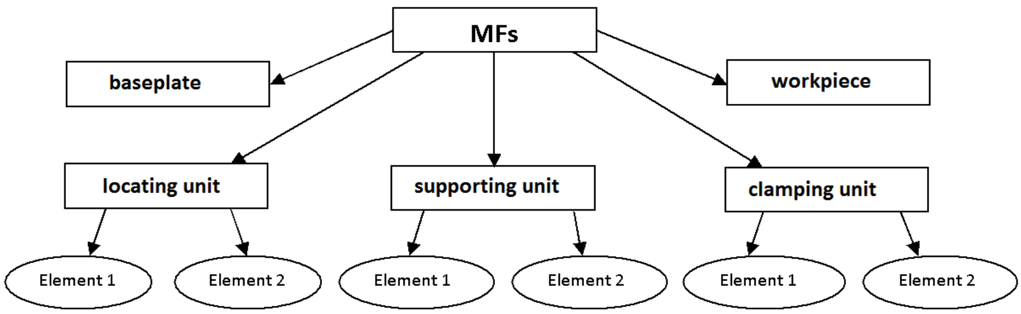


Figure 3-23. Modular fixture structure to assemble a workpiece to a baseplate [44].

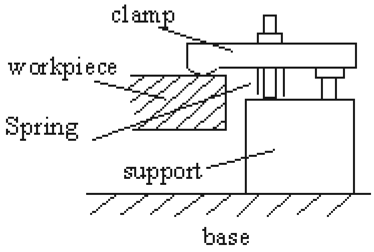


Figure 3-24. Typical sample of a fixture unit [45].

This unit consists of a supporting element and clamping element for performing a clamping function on the workpiece, and this functional unit is connected to the baseplate. Moreover, the structure of the fixture elements and the functional units can be represented by using a scenegraph data structure. A scenegraph represents the elements as nodes, and the edges are used to describe the relationship between the nodes [46]. In addition, the position of each node in the environment can be described as a hierarchical structure by using the scenegraph [43]. A sample of a hierarchical structure of an assembly model is shown in **Figure 3-25**.

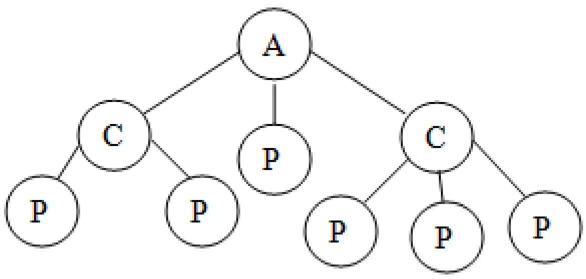


Figure 3-25. Scenegraph hierarchical assembly model [43]. Where A is a fixture assembly, C is a functional unit and P is a fixture element.

3.4 Visual Basic Fundamentals

Visual Basic (VB) is a programming language developed by Microsoft. This language has been used widely among a high percentage of developers as a “primary development tool”, compared to other programming languages [47]. VB is the advanced version of the BASIC language with an “Integrated Development Environment (IDE)” which makes the program more efficient for creating, running and debugging operations [48]. VB supports many features and functions such as Win32 API accessing and graphical user interfaces [48]. Moreover, VB is considered the engine for the building of macros in all Microsoft software [47]. Therefore VB has become an important tool for building different programs for many applications. **Figure 3-26** illustrates the VB screen accessed in order to create new project. As for many applications, existing and recent projects can be also opened and modified. There are different types of projects that can be generated in VB. For simple programming purposes, Standard EXE is most commonly used by programmers. For more advanced programming functions, ActiveX projects are used.

In this research, an ActiveX DLL (Active X dynamic link libraries) has been created. This project allows the programmer to integrate VB with different Windows applications. Furthermore, this project controls the features and operations of other applications by creating new menus and toolbars in the applications environment. For database management purposes, VB is the engine of Microsoft Access for the building of the database, and this gives the programmer the opportunity to efficiently control the database [47].

Figure 3-26. Visual Basic new project screen.

3.5 SolidWorksAPI

Application programming interface (API) is a tool to write code in a programming language within another application. As a result, a direct integration between different applications can be developed [49]. SolidWorks is one of the applications which support API with different programming languages such as C++, Visual Basic and Visual Studio. SolidWorks API automates the design and assembly operations by creating codes in a specific programming language, and it has been applied in different design methods [49]. API is used to develop a web service material database with SolidWorks for simplifying the selection of materials by the designer [50]. Bo, Qin and Fang developed a standard parts library by using Visual Basic code with SolidWorks API functions [51]. This system is based on creating Visual Basic forms for improving the design efficiency. Peng, Jing and Xiaoyan applied Visual Basic Net in second development of SolidWorks for the simulation of a "3D module of an architectural process" [52]. They generated an add-in VB project for Solidworks for automating the assembly process. SolidWorks API was employed in designing a model of a centrifugal fan impeller [53]. This method is based on geometric features to create the second development by SolidWorks and VB 6. Moreover, API with a user interface employing a knowledge-based design features application can help customise CAD systems for certain tasks [54]. Reuse software was developed by applying SolidWorks API by Tian and Liu [55]. The system was built using VB for the secondary development of the CAD system for a standard part. Zhen and Yingyi introduced an assembly method based on SolidWorks [56]. They illustrated the steps in the assembly automation procedure by using Visual C++. They also explained how the information on the parts was stored in the database. The secondary development of SolidWorks was used by Yang for developing an intelligent system for an assembly based on a parametric design [57]. Delphi programming language, SolidWorks API and an Access database were the techniques used for creating this system.

3.6 Summary

In order to correctly perform the machining operations for a specific workpiece, it is important to define the proper locating and clamping methods. This is based on the specifications and the shape of the workpiece and the machining features. A modular fixturing concept has been applied in manufacturing systems since its benefits have been recognised. These benefits result in a reduction in cost and time of production. By employing SolidWorks API functions with VB programming language, an integrated modular fixture system was developed for a semi-circular workpiece in this research. Side clamping was used as the clamping method and V-block as a locating method. The integrated system is divided into two main stages, the selection process of fixture elements (Chapter 4) and the assembly automation (Chapter 5).

Chapter Four

4. The Selection Approach for Modular Fixture Elements

This Chapter introduces the approach of selecting MF elements. This is the first stage of the integrated modular fixture system developed in this study. In this Chapter, the knowledge base for MF is developed and IF-THEN rules are created for a semi-circular workpiece. The knowledge base includes knowledge required for determining the locating method, the clamping method and the appropriate fixture elements. The VisiRule expert system toolkit is employed to develop the decision process for the selection approach. The results of this process are implemented to simulate the assembly process of MF in Chapter 5.

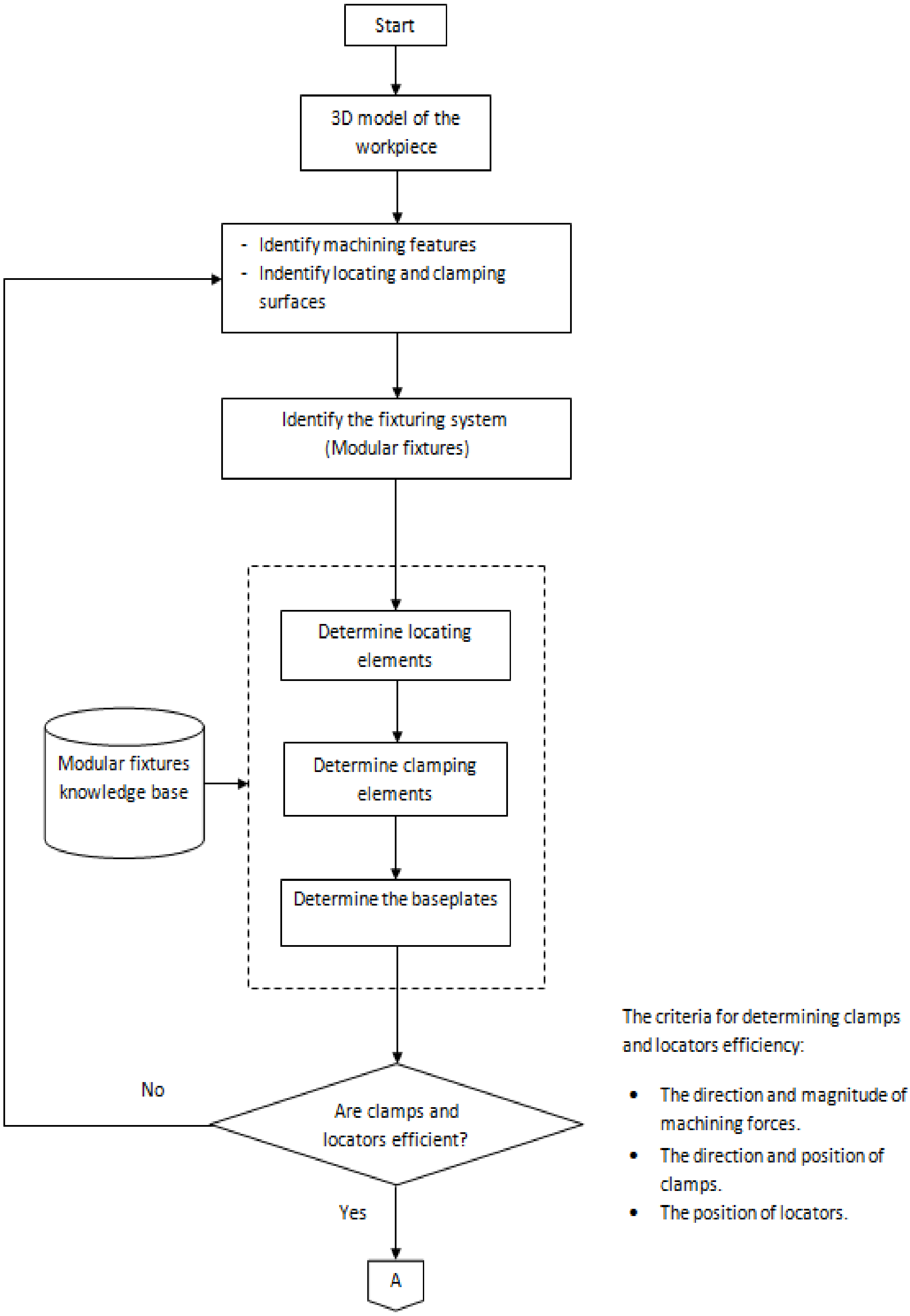
4.1 The Integrated System

The working mechanism of the integrated system is presented in **Figure 4-1**. Starting with the design of a 3D model of the chosen workpiece, the machining features and the locating/clamping surfaces are determined. After that the geometric specifications, shape of the workpiece and nature of fixturing are identified. In this research, semi-circular workpieces and MF system are considered.

As shown in **Figure 4-1**, the first stage of the system is the selection of the locating method, clamping method, baseplate, locators and clamps. For this purpose a knowledge base is built to meet the requirements of the selection process. After completing this stage, the efficiency of the clamps and the locators needs to be checked by considering three main criteria; the direction and magnitude of the machining forces (cutting forces), the direction and the position of clamps, and the position of locators. Once the efficiency of the selection is verified, the next stage is to assemble the fixture elements in the SolidWorks environment.

Two important components are needed in this stage, the assembly knowledge base, and the feature library of the fixture elements. The assembly knowledge base contains the necessary relationships for assembling the fixture elements together in order to properly complete the fixture layout. This knowledge base is explained in Chapter 5. The feature library contains the standard modular fixture components with their details, and is explained in Chapter 6. By completing these two components, the workpiece, the locators, the baseplate, the clamps and the other fixture elements are inserted and assembled in the assembly environment. The process is automated by using SolidWorks API and Visual Basic 6 which will be explained in Chapter 5.

The fixturing layout is generated after completing this stage, and this layout also needs to be approved. It is important to determine that there are no collisions between the fixture elements and the workpiece, and that all components are perfectly located in their positions. The interference detection tool in SolidWorks is applied for this purpose, and in case any collision is found the assembly process needs to be repeated until it meets the requirements of an acceptable layout. This integrated system can be applied to different kinds of workpieces and fixturing systems. For clamping semi-circular workpieces in this work, V-blocks are used as locating elements together with side clamps. Use of side clamping frees the top face of the workpieces for machining processes. For other workpiece shapes and fixturing configurations, the knowledge base for the selection and assembly processes needs to be modified in order to generate the appropriate fixturing layouts that meet the machining requirements.


Start
3D model of the workpiece
- Identify machining features
- Indentify locating and clamping surfaces
Identify the fixturing system (Modular fixtures)
Determine locating elements
Modular fixtures knowledge base
Determine clamping elements
Determine the baseplates
No
Are clamps and locators efficient?
Yes
A
The criteria for determining clamps and locators efficiency:
• The direction and magnitude of machining forces.
• The direction and position of clamps.
• The position of locators.

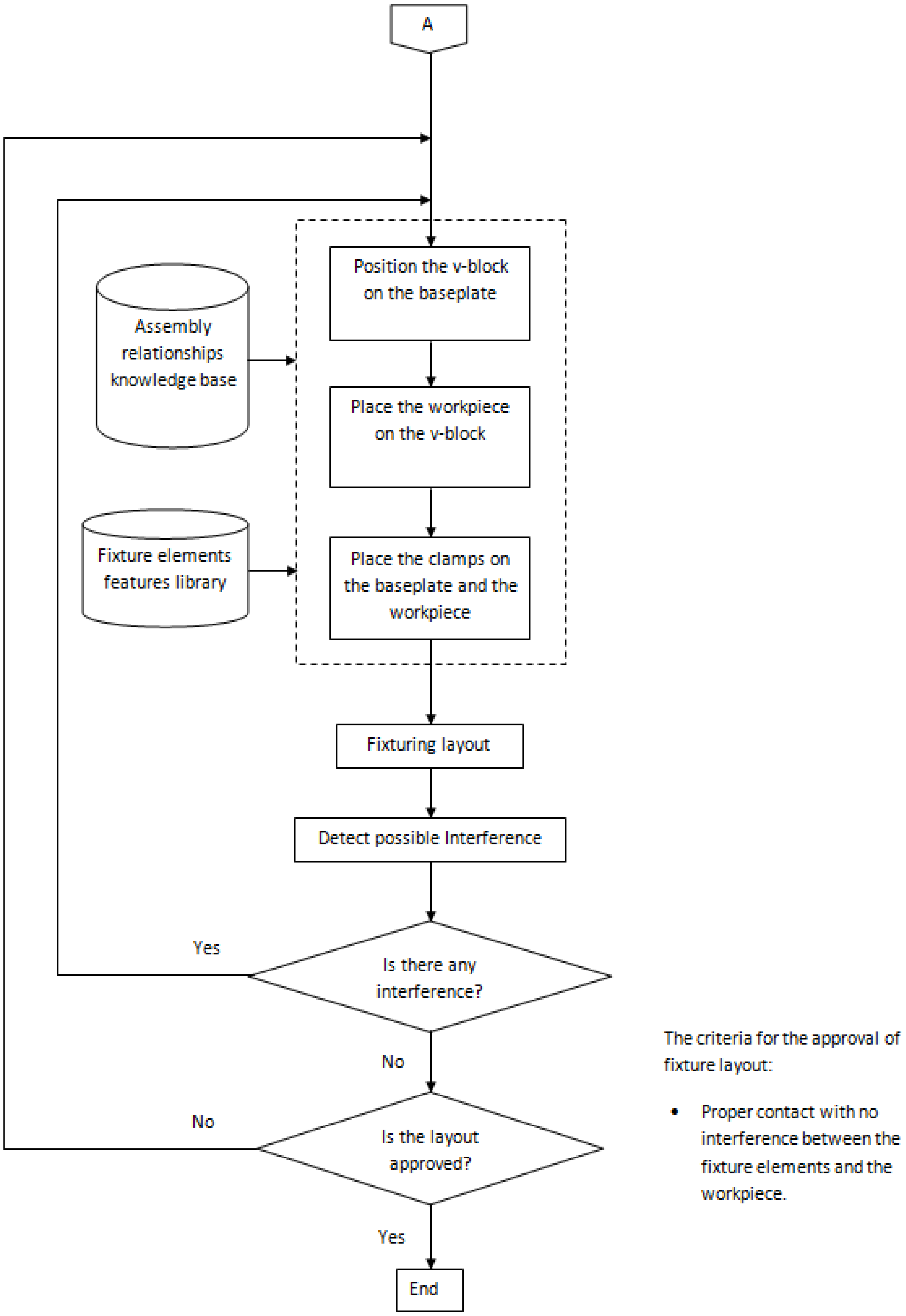


Figure 4-1. The working mechanism of the integrated system developed.

4.2 Knowledge Base Development for the Selection Process

The expertise knowledge for determining the appropriate fixturing method and the related fixture elements for the given workpiece is stored in the knowledge base in an IF-THEN rule format and divided into four groups. The rules can be found in Appendix 1 **.** The first group contains five rules for identifying the requirements for the machining features of the workpiece, the fixturing system applied, and the workpiece surfaces needed for locating and clamping purposes. An example of these rules is:

Rule 002

If the workpiece is semi-circular and a modular fixture design is used, *then* the active locating surface is defined. This kind of workpieces has four surfaces; rectangular surface, semi-cylindrical surface and two semi-circular surfaces. Defining the active locating surface is important for determining the locating method and completing the rest of the process.

The pesudocode for this rule is:

If *workpiece = semi-circular* ***AND*** *MFs are used*

Then *define the active locating surface*

End If

The programming VB code is:

```
Dim swApp As Object
Dim MFs As Object
Dim Workpiece As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swApp =Application.SldWorks
Set Workiece = semi-circular
boolstatus = Workpiece.Activesurface("C:\Users\Desktop\Workpiece)
End Sub
```

The second group contains five rules for determining the locating method and their fixture elements. An example of these rules is:

Rule 006

If the workpiece's radius is more than or equal to 9 mm *and* smaller *than* 25.4 mm, then small mini V-blocks are used. These V-blocks are appropriate for this range of diameters, and have two holes for mounting on the baseplate.

The pseudocode for this rule is:

If *Radius = > 9 mm* ***AND****< 25.4 mm*

Then *V-block = small mini*

End If

The programming VB code is:

```
Dim radius As Object
Dim V-block As Object
Dim Workpiece As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set Workiece = semi-circular
boolstatus = Workpiece.Radius (=> 9, < 25.4)
Set V-block = small mini
End Sub
```

The third group identifies the clamping method and the clamping elements with ten rules. An example of these rules is:

Rule 010

If the workpiece is semi-circular and the rectangular surface is machined, *then* the machined area of this surface is calculated. This is important to define the clamping method.

If the whole rectangular surface is machined, *then* side clamping is used. This clamping method allows the workpiece to be held whilst keeping the surface clear for machining. If the machined area is equal to 50% or less than the whole area of the rectangular surface, then top clamping could be used. This depends on the shape and location of the pattern to be machined on the surface.

The pseudocode for this rule is:

***If** machined area = 100% rectangular surface*

***Then** Side Clamping*

Else Top Clamping

End If

The programming VB code is:

```
Dim Machined area As Object
Dim rectangular surface As Object
Dim Workpiece As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set machined area = 100%
Set clamping = Side clamping
End Sub
```

The last group contains ten rules for determining the baseplate. An example of these rules is:

Rule 023

If the machining area is less than or equal to 0.094 m^2, *and* standard V-blocks are used, *then* a 400 mm standard baseplate is to be used.

The pseudocode is:

***If** machining area = 0.09 m^2 **AND** Locating = V-block*

***Then** Baseplate = 400 mm*

End If

The programming VB code is:

```
Dim machining area As Object
Dim V-block As Object
Dim Workpiece As Object
Dim Locating As Object
Dim Baseplate As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set Workiece = semi-circular
Set locating = V-block
Set Machining area = 0.09
boolstatus = Baseplate.Add ( 400 mm)
End Sub
```

The knowledge base is used along with the VisiRule expert system to generate the code in Flex programming language as explained in the next section.

4.3 Expert System Implementation (VisiRule Toolkit)

The selection process of MF elements in this research consists of three main stages:

1. Determine the locating method and the related locating elements;
2. determine the clamping method and the related clamping elements; and
3. determine the baseplate.

This process is driven by the VisiRule expert system. VisiRule is a powerful tool which allows drawing of rules in the charts form in order to model the decision process. Different formats can be used when developing the charts, such as an equation format, statement format and rules format [58]. The results of this process are the knowledge base for the fixture element selection as a programming code in a Flex language format. The Flex format can be converted into other programming languages and used for several applications. After defining the workpiece and the kind of fixturing system, the selection of the fixture elements commences.

4.3.1 Determining the Locating Method and Related Elements

The flowchart for the process of locating is shown in **Figure 4-2**. The VisiRule flowchart developed for this process is in Appendix 2**.** The format of the VisiRule flowcharts is based on the equations used to identify the appropriate elements. The process starts with defining the kind of workpiece. In this study a semi-circular workpiece has been chosen to develop the appropriate selection approach. After determining the workpiece, the fixturing system should be identified, and in this research an MF system is to be used with standard fixture elements.

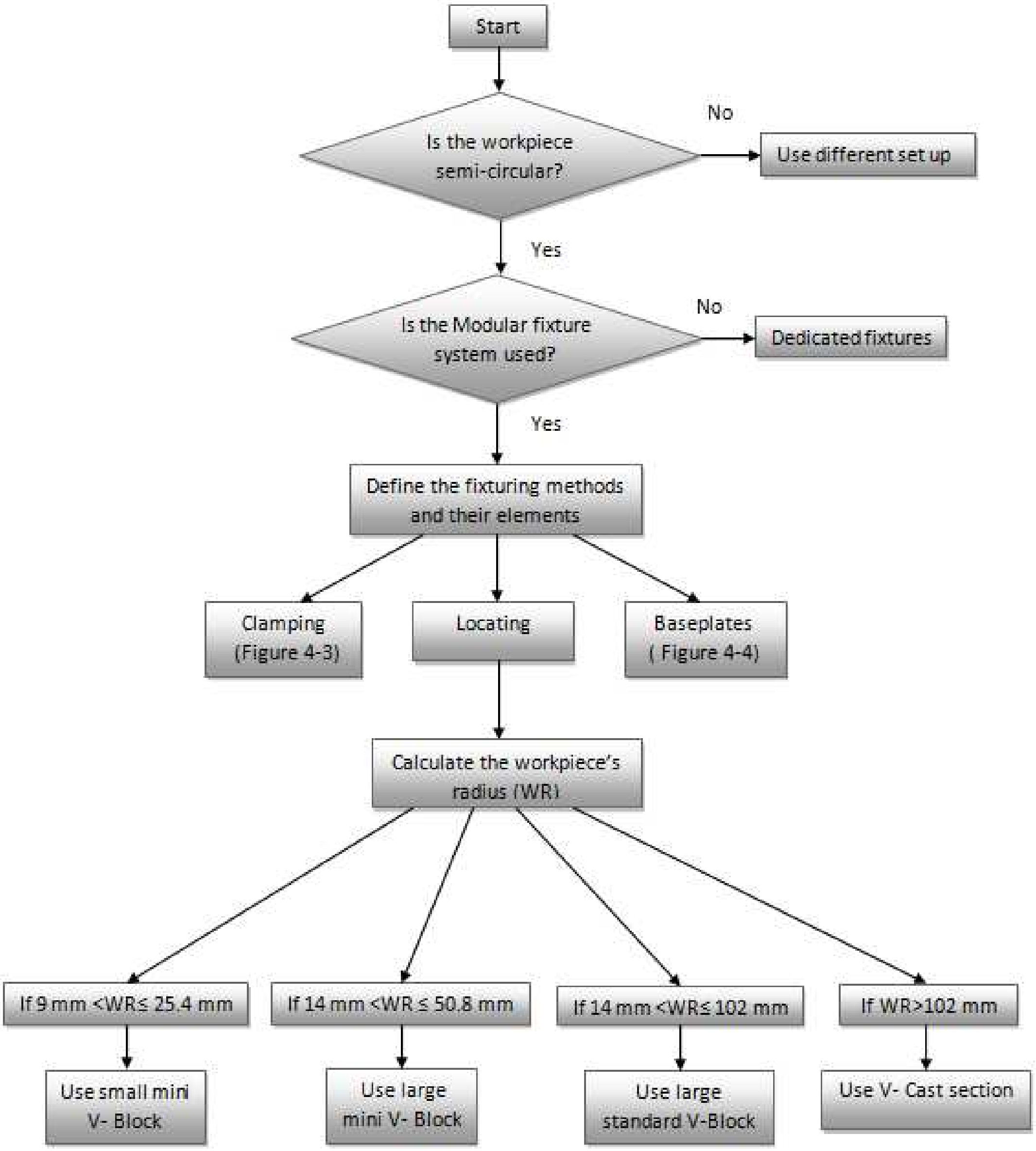


Figure 4-2. The mechanism for determining the location method and locating elements.

The main criterion for locating is the workpiece radius. Defining the radius allows the locating elements to be specified, which in this work are V-blocks. Four radius categories are considered for locating selection, a radius between 9 mm and 25.4 mm, a radius between 14 mm and 50.8 mm, a radius between 14 mm and 102 mm, and a radius of more than 102 mm. For these four categories four V-blocks are selected; small mini V-block, large mini V-block, large standard V-block and V-cast section, respectively.

4.3.2 Determining the Clamping Method and Related Elements

After defining the locating method and its elements, the clamping method and its elements are defined (**Figure 4-3**). The VisiRule chart for clamping is presented in Appendix 3. The clamping system used in this work is side clamping in order to keep the top surface of the workpiece (the rectangular surface) clear for the machining operations, and the clamping components are pivoting edge clamp, downthrust backstop, and riser blocks. The main criterion for clamping is the height of the workpiece, and this is important in order to define the kind of clamps and other clamping components required. Clamping selection starts by defining the clamping system. If side clamping is used then the kind of clamps to be used is identified; in this study pivoting edge clamps have been chosen.

Next the height of the workpiece is identified. If the workpiece's radius is taken as the height, then the process continues with the determination of the clamping components. However, if another dimension is taken as the height, another clamping setup could be employed by considering other types of clamping components. Three height ranges are found for the selected workpiece, less than or equal to 100 mm, more than 100 mm and less than or equal to 125 mm, and more than 125 mm and less than or equal to 150 mm. For the first range, no riser block is needed, while for the other two ranges 25.4 mm and 50.8 mm riser blocks are needed. Two categories can be used for the pivoting clamp and downthrust backstop, standard and heavy. This depends on whether the category of the baseplate is standard or heavy. Standard and heavy categories are related to the fixture element specifications from the CarrLane catalogues [59].

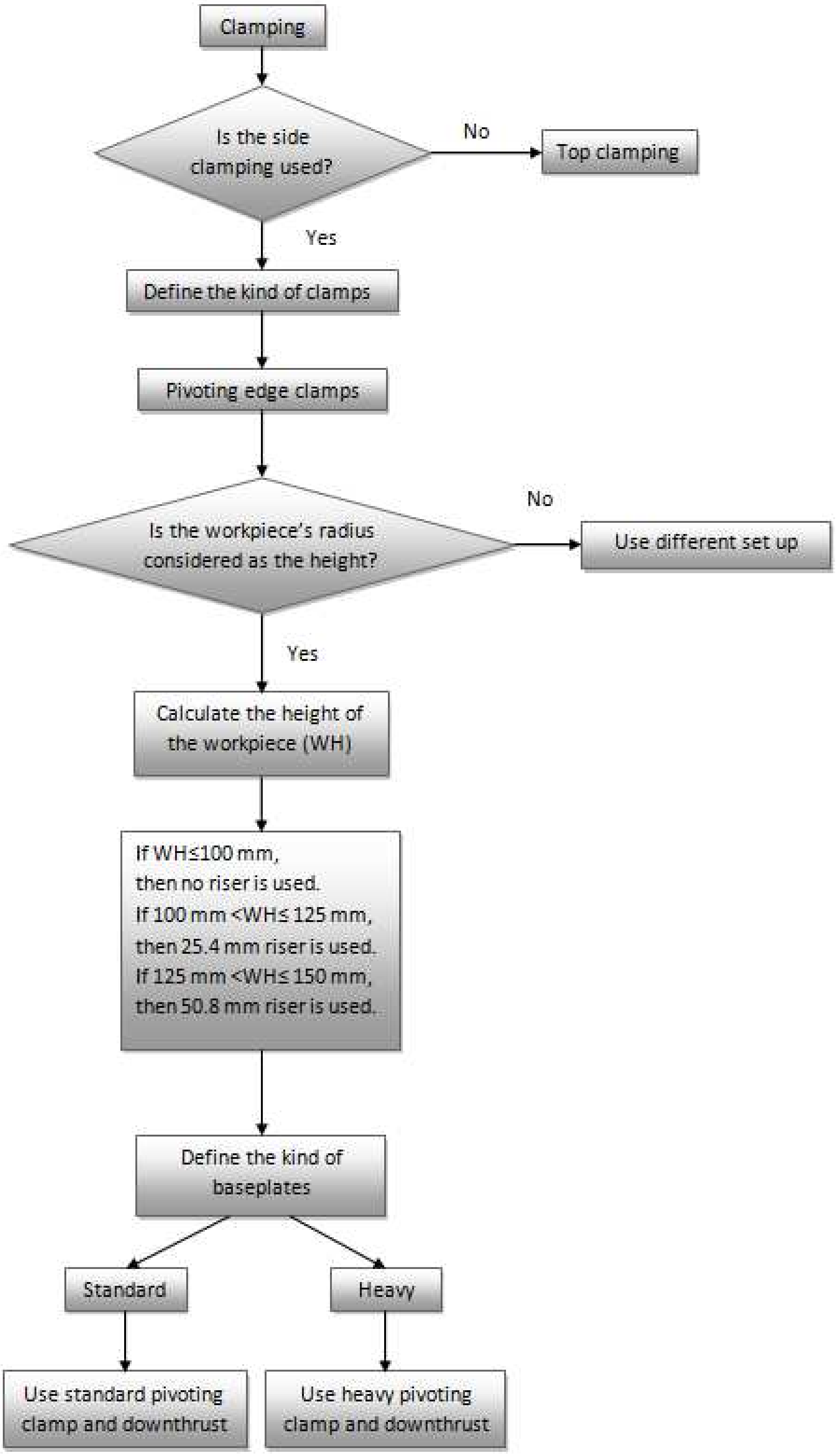


Figure 4-3. The mechanism for determining the clamping method and kind of clamps.

4.3.3 Determining the Baseplate

In order to define the baseplate it is important to specify which MF system has to be used. In this research, hole-dowel MF is used, and **Figure 4-4** shows the process of the baseplate selection process. It is also necessary to determine the clamping method, the clamps and the type of V-blocks. Moreover, the workpiece machining area should be calculated to determine the size of the baseplate. Three classes, mini, standard, and heavy, with four sizes of the baseplate, 400mm, 500mm, 630mm, and 800mm, are considered. The VisiRule chart developed for this process is presented in Appendix 4.

As shown in **Figure 4-4**, side clamping is used with a pivoting edge clamp beside the hole-dowel system. In terms of calculating the machining area (A), four ranges are used for the semi-circular workpiece, an area less than or equal to 0.094 m^2, more than 0.094 m^2 and less than or equal to 0.0146 m^2, more than 0.0146 m^2 and less than or equal to 0.0232 m^2, and an area more than 0.0232 m^2 and less than or equal to 0.0037 m^2. Two classes of baseplate can be used for each range of the machining area, 400 mm mini and standard classes for the first area, 500 mm standard and heavy classes for the second area, 630 mm standard and heavy classes for the third area, and 800 mm standard and heavy classes for the fourth area, respectively. This is dependent on the V-blocks' class, which should be compatible with that of the baseplate.

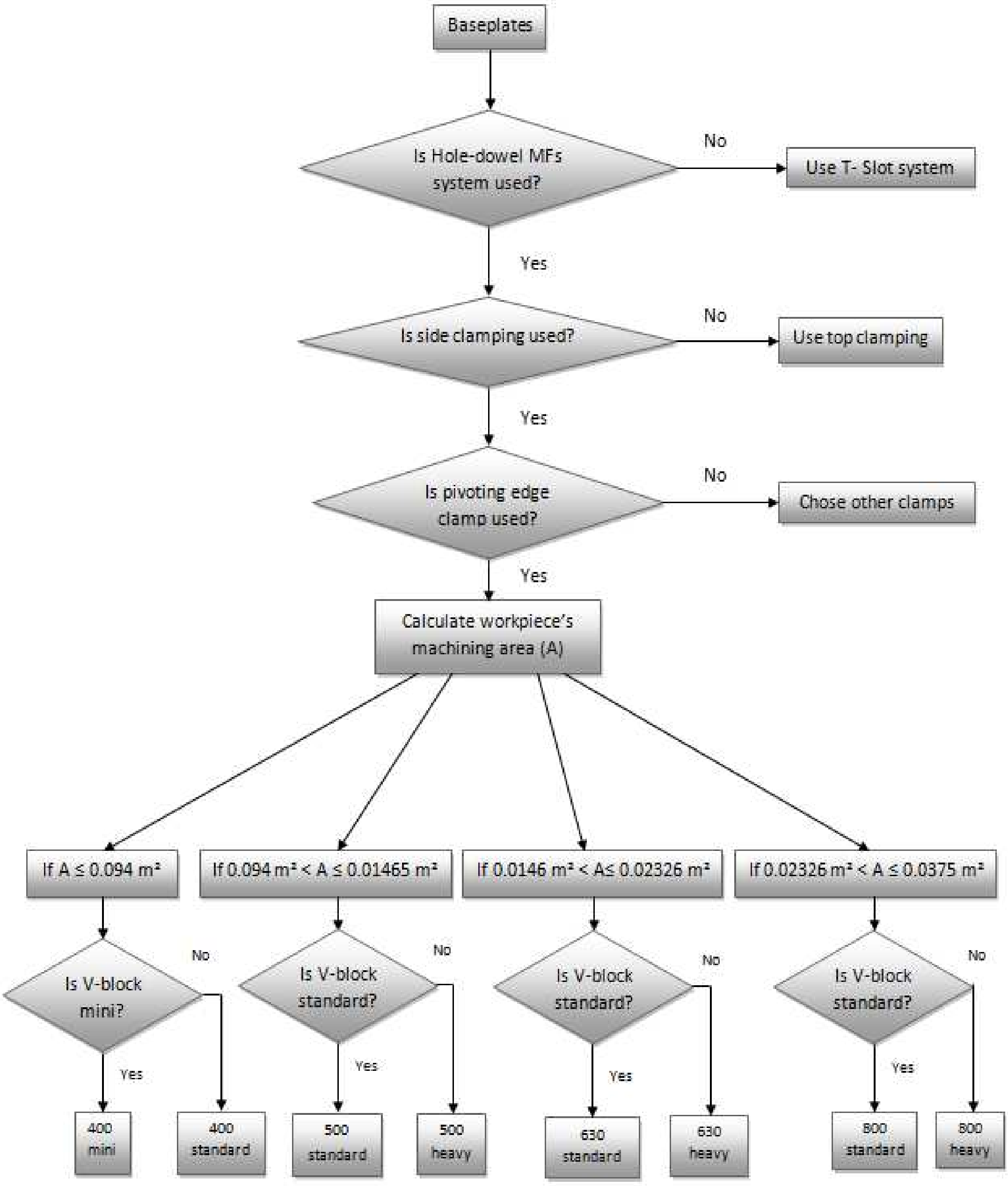


Figure 4-4. The mechanism for determining the kind of the baseplate. (A) refers to the machining area.

4.4 Results and Discussion

The knowledge base developed in this Chapter is employed to create flowcharts in VisiRule. The question method is applied to create these charts, as shown in **Figure 4-5**.

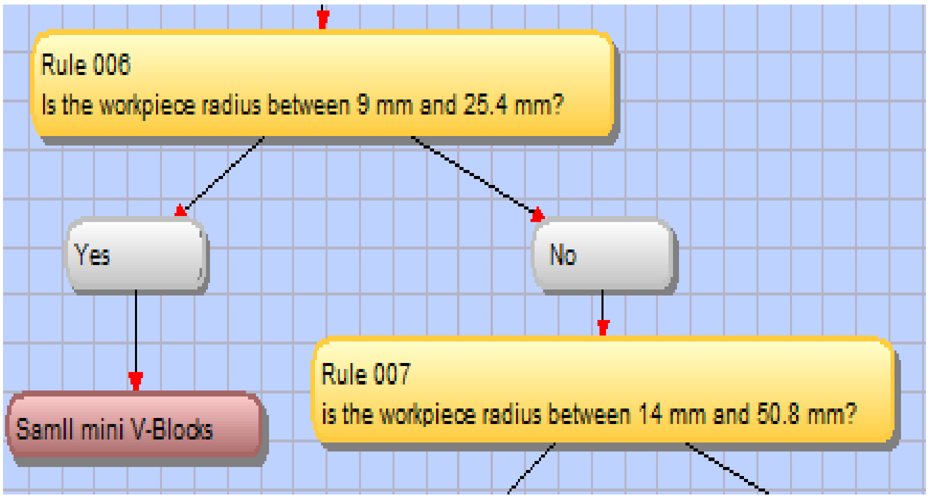


Figure 4-5. A section of the question-based method flowcharts used in VisiRule.

By completing the charts, the programming code of this process is generated, and the interface of the software is shown in **Figure 4-6**. From this interface, by clicking the Run command button another window is opened and the software interacts with the user as shown in **Figure 4-7**. The generated code (Appendix 5) is based on the Flex programming language. This code can be used with other programming languages such as VB. In this study, the Flex code is transferred to XML format to make it readable by VB. After that, VB is integrated with SolidWorks in order to obtain the results from VisiRule and to simulate the assembly process of the fixture elements. This integration is explained in detail in Chapter 5.The selection approach presented in this Chapter could be improved by directly integrating the expert system toolkit with SolidWorks. This would allow the user to run the selection process within SolidWorks and select the proper elements for a certain workpiece in order to insert them in the assembly environment.

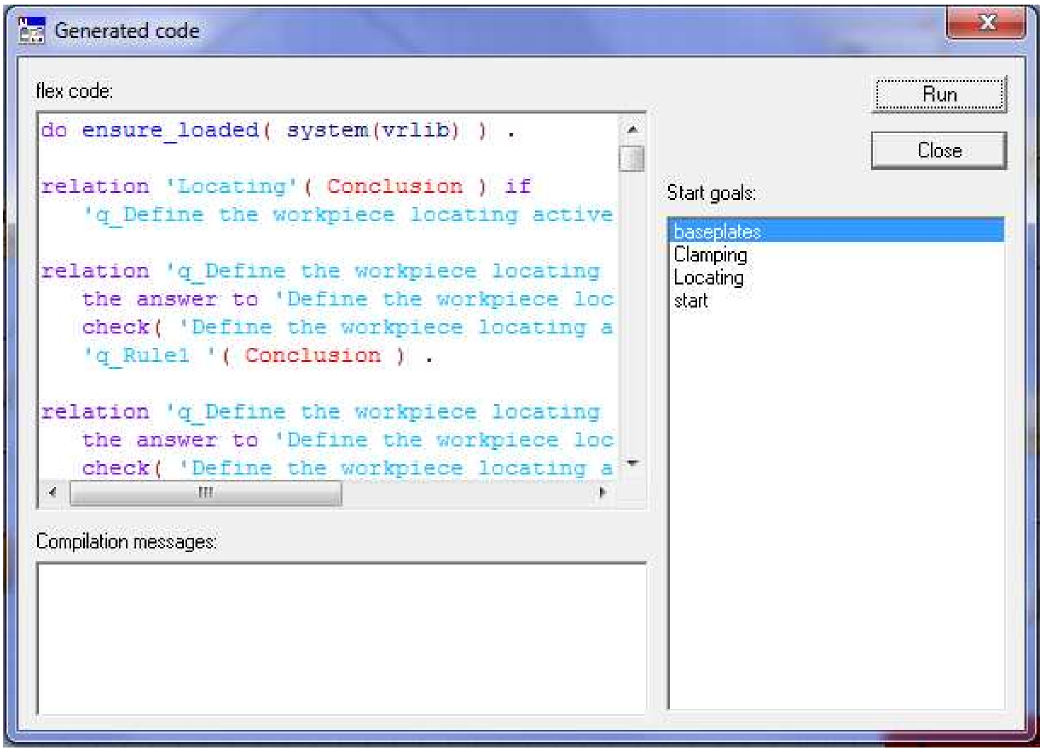


Figure 4-6. The interface for the generated code in VisiRule.

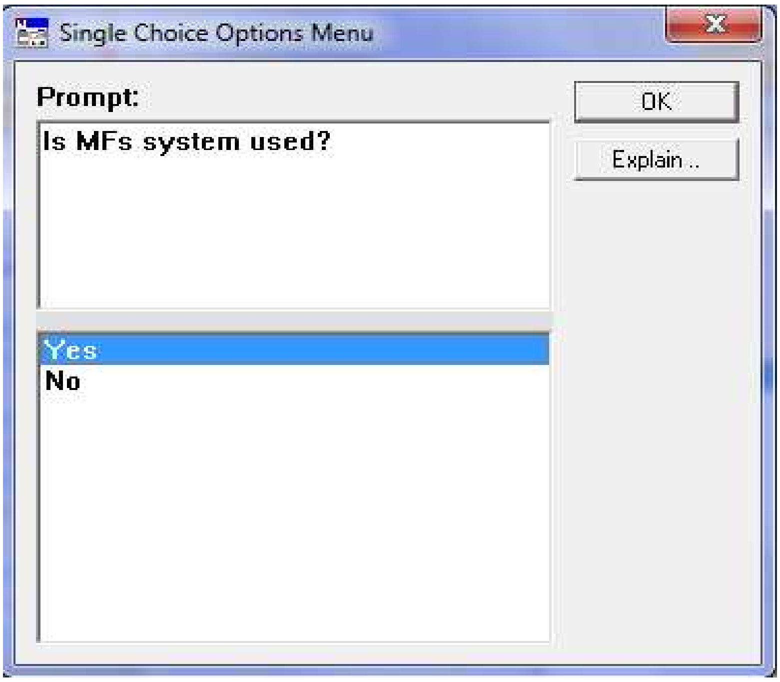


Figure 4-7. The interface for the options in VisiRule.

4.5 Summary

This Chapter explained the developed development of the code for the selection approach for modular fixture elements. This code is generated through the flowchart concept in VisiRule. The code can be transferred to other programming languages and can be read by VB, which in this research is integrated with SolidWorks. The knowledge base of MF is developed, including the requirements of choosing the proper locating/clamping methods and baseplates for a semi-circular workpiece. Machining and geometry features are considered for defining the appropriate fixture elements. Defining the proper fixture elements opens the way to the next stage, the simulation and automation of the MF.

Chapter Five

5. The Automation and Simulation of Modular Fixture Assembly

This Chapter introduces the assembly process for MFs design, based on the SolidWorks assembly mating features. Each step, from the insertion of the elements into SolidWorks through to their assembly, is explained in detail. The automation includes creating new menus for adding the selected elements and for assembling them. Macro functions in SolidWorks were used for simulating the automation process for fixture element design and assembly. The proper mating features and their format were included, and the knowledge base for this process has been developed. Based on the knowledge base, SolidWorks API is implemented with a VB ActiveX project for automating the MF assembly process.

5.1 The Assembly Process for the Developed System

The selection process for fixture elements was completed using the expert system as described in Section 4.1. For a semi-circular workpiece, the assembly process is completed by using SolidWorks in a CAD environment. The flowchart of this process is shown in **Figure 5-1** . The assembly process is described in the steps below:

Step 1: A new assembly document is opened in the SolidWorks environment.

Step 2: The fixture elements are inserted into the SolidWorks environment by choosing the **Insert** menu from the pull down menu toolbar. This menu allows inserting Existing Part/Assembly or New Part, and New Assembly.

In this research, the fixture elements could be retrieved from the CarrLane's website as SolidWorks files and stored prior to being inserted in a new assembly environment. The fixture elements can be inserted using the **Browse** button in the property manager.

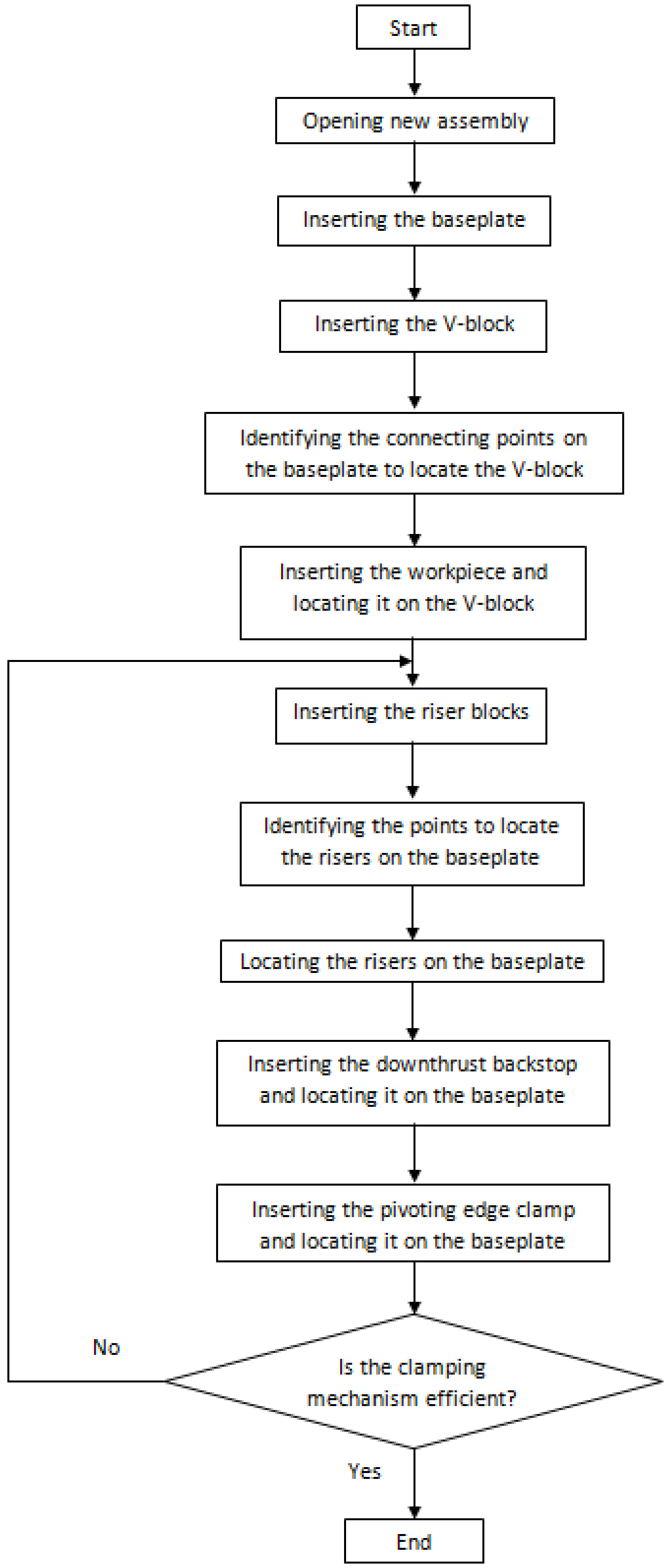


Figure 5-1. The flowchart for the assembly process in the 3D SolidWorks environment.

Step 3: The first fixture element inserted is the baseplate. In the SolidWorks assembly environment, the first inserted component will be the base for the rest of the components. Thus this element cannot be moved or rotated. A MF400804 baseplate of 800 mm x 800 mm in size, standard class, with a 14 x14 grid pattern is used in this study (**Figure 5-2**). The specification of this baseplate can be found in Appendix 6.

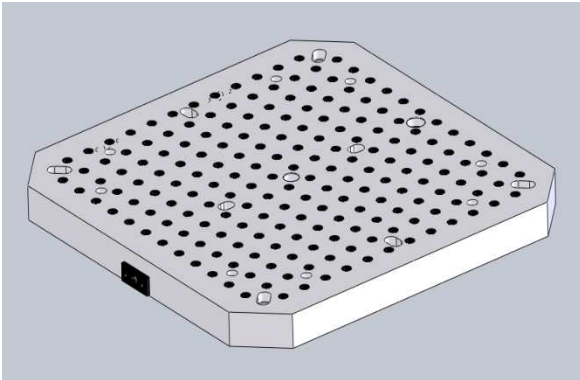


Figure 5-2. The MF400804 baseplate with 14x14 pattern and 800 mm x 800 mm size.

Step 4: The second fixture element to be inserted is the V-block or V-cast section (**Figure 5-3**). This V-block was designed to meet the size requirement of the workpiece chosen in this research. The specifications of the V-block can be found in Appendix 7.

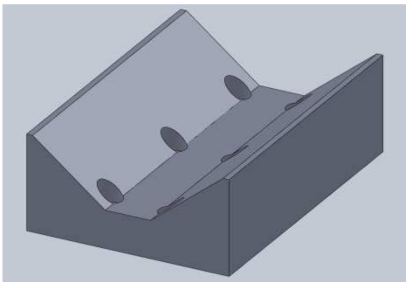


Figure 5-3. The designed V-block (cast section).

Step 5: The V-block is attached to the baseplate. For assembly, the following mating features for the selected faces of the fixture elements were considered:

- Coincident mate between the top face of the baseplate and bottom face of the V-block.

- Parallel mate between the upper long edge of the V-block and the opposite edge of the baseplate.

- Concentric mate between the 7x6 hole of the baseplate's grid pattern and one of the holes of the V-block as defined by the designer to meet the correct location of the V-block on the baseplate.

The formats of these mates in SolidWorks are:

Conceentric1(MF400804<1>, V-block<1>)

Coincident1(MF400804<1>, V-block<1>)

Parallel1(MF400804<1>, V-block<1>)

In which MF400804 is the name of the baseplate and the number (currently 1) after each mate refers to the sequence of the mating feature in the assembly procedure. The number (currently 1) after the element's name refers to the number of the elements involved in the mating procedure

Step 6: The workpiece is inserted into SolidWorks. The specifications of the workpiece can be found in Appendix 8.

Step 7: The workpiece is located on the V-block by considering its locating active surface that was defined from the selection process in section 4.2, which in this workpiece is the external semi-cylindrical surface (**Figure 5-4**). To do so, the following mates are followed:

- Tangent mate number one between the workpiece's semi-cylindrical surface and V surface number one of the V-block.

- Tangent mate number two between the workpiece's semi-cylindrical surface and V surface number two of the V-block.

- Distance mate between one of the workpiece's side surfaces and one of the opposite faces of the V-block to correctly locate the workpiece on the V-block.

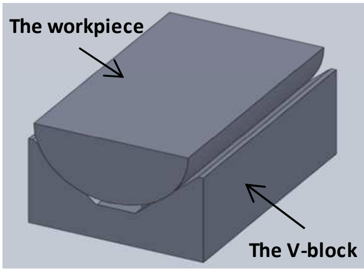


Figure 5-4. The assembly of the workpiece and the V-block.

The formats of the mating features in SolidWorks are as follows:

Tangent 1(Part<1>, V-block<1>)

Tangent 2(Part<1>, V-block<1>)

Distance 1(Part<1>, V-block<1>)

Two tangent mates and one distance mate are used for one part and one V-block. **Part** is the name of the chosen workpiece.

Step 8: The riser block is inserted. Two riser blocks are used in this work and their specifications are given in Appendix 9.

Step 9: The risers are located on the baseplate and the following mates are followed:

- Concentric mate between the 7x2 hole of the baseplate's grid pattern and one of the holes of the riser blocks as defined by the designer to meet the correct location.

- Coincident mate between top face of the baseplate and the bottom face of the riser.

- Parallel mate between one of the riser's edges and the opposite edge of the baseplate.

The formats of the mates for the first riser are:

Concentric 2(MF400804<1>, MF401601<1>)

Coincident 3(MF400804<1>, MF401601<1>)

Parallel 2(MF400804<1>, MF401601<1>)

The formats of the mates for the second riser are:

Concentric 3(MF400804<1>, MF401601<1>)

Coincident 4(MF400804<1>, MF401601<1>)

Parallel 3(MF400804<1>, MF401601<1>)

Where MF400804 and MF401601 are the names of the baseplate and the riser, respectively, and the numbers after the mates refer to their sequence in the assembly procedure.

Step 10: The downthrust backstop is inserted (**Figure 5-5**), and its specification is given in Appendix 10.

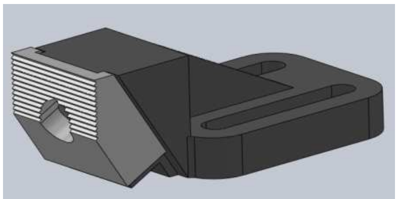


Figure 5-5. The downthrust backstop used in this study.

Step 11: The downthrust backstop is located on the riser and assembled onto the workpiece. The following mates are followed:

- Coincident mate between the bottom face of the downthrust backstop and the top face of the riser.

- Distance mate between the long edge of the downthrust and the opposite edge of the riser to correctly locate the downthrust.

- Coincident mate between the JAW of the downthrust and one of the workpiece's side faces.

- Parallel mate between the same edge of the downthrust, which is used for the distance mate, and the opposite edge of the baseplate to prevent the sideways movement of the downthrust.

The formats of the mates in SolidWorks are:

Coincident 5(MF401101<1>, MF401601<1>)

Distance 2(MF401101<1>, MF401601<1>)

Coincident 6(MF401101<1>, Part<1>)

Parallel 4(MF401101<1>, MF400804<1>)

Where MF401101 is the name given to the downthrust backstop.

Step 12: The pivoting edge clamp is inserted (**Figure 5-6**), and its specifications are in Appendix 11.

Step 13: The pivoting clamp is located on riser number two and assembled onto the workpiece. The following mates were used:

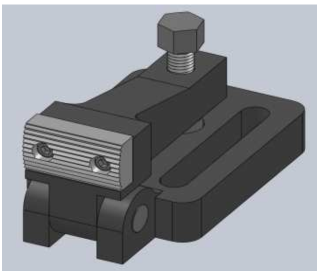


Figure 5-6. The pivoting clamp used in step 13.

- Coincident mate between the bottom face of the clamp and the top face of riser number two.

- Distance mate between the long edge of the clamp and the opposite edge of the riser as defined by the designer to correctly locate the clamp.

- Parallel mate between the long edge of the clamp and the opposite edge of the baseplate to prevent sideways movement of the clamp.

- Coincident mate between the clamps's JAW and the workpiece's side face number two in order to hold the clamp against the downthrust backstop.

The formats of the mates in SolidWorks are:

Coincident 7(MF401001<1>, MF401601<1>)

Distance 3(MF401001<1>, MF401601<1>)

Parallel 5(MF401001<1>, MF400804<1>)

Coincident 8(MF401001, Part<1>)

In which MF401001 is the name given to the pivoting edge clamp.

5.2 The Assembly Knowledge Base (Side Clamping)

A knowledge base is developed for side clamping. This knowledge base is based on the assembly process explained above. The rules of this knowledge base are as follows:

Rule 1

If the locating method is V-blocks, then define its locating surface on the baseplate. The locating surface is the bottom face of the V-block.

Rule 2

If the locating surface of the V-block is defined, then identify the location of the V-block on the baseplate. This depends on the sizes of the V-block and baseplate. The location can be defined by calculating the distance of the holes on the baseplate for the correct location.

Rule 3

If the surface is defined and the location is calculated, then use a coincident mate to locate the V-block on the baseplate.

Rule 4

If the surface is defined and the location is calculated, then use a concentric mate to assemble the V-block with the baseplate.

Rule 5

If the V-block is located, then use a tangent mate between the V-block and the baseplate to prevent the V-block rotating.

Rule 6

If the workpiece is semi-circular, and the semi-cylindrical surface is used for locating, use a tangent mate to locate the workpiece on the V-shaped surfaces of the V-block.

Rule 7

If a riser is used, then define the location of the riser by calculating the distance on the baseplate to locate the riser in the correct position.

Rule 8

If the riser’s location is defined, then use a coincident mate to locate the riser on the baseplate.

Rule 9

If the riser's location is defined, then use a concentric mate to assemble the riser with the baseplate.

Rue10

If a pivoting clamp is used, and the workpiece is semi-circular with V-block locating, then define the surface of the workpiece for clamping. In this research, the lateral surfaces of the workpiece are used for clamping.

Rule 11

If the clamping surfaces are defined, then use a coincident mate to locate the clamp on the riser.

Rule 12

If the clamping surfaces are defined, then use a coincident mate to assemble the clamp with the workpiece.

Rule 13

If a downthrust backstop is used, then define the other side surface of the workpiece to complete the clamping mechanism.

Rule 14

If a downthrust backstop is used, and the surface is defined, then use a coincident mate to locate the downthrust on the riser.

Rule 15

If a downthrust backstop is used, and the surface is defined, then use a coincident mate to assemble the downthrust backstop on the workpiece.

5.3 Add-in Project Development

The results of the selection process and the knowledge base of the assembly process are used for creating add-in code in order to build menus in the SolidWorks environment for automating the assembly process. The menus are built by using VB integrated with SolidWorks API (Application Programming Interface). In this research, an ActiveX DLL project in VB is developed, incorporating SolidWorks libraries as references. In this case, two libraries should be referenced; these are the **SldWorks Type** Library and **SolidWorks Exposed Type** library. Adding these libraries to the ActiveX DLL project allows SolidWorks commands and functions to be controlled. Below is an example of the add-in code developed:

```
Dim axSldWorks As SldWorks.SldWorks
Dim axCookie As Long 'holds value created in SwAddin_ConnectToSW
Dim axToolbarID As Long 'toolbar ID if toolbars used
Dim axActiveDoc As SldWorks.ModelDoc2
Dim axTargetDoc As SldWorks.ModelDoc2
Private Function SwAddin_ConnectToSW(ByVal ThisSW As Object, ByVal Cookie
As Long) As Boolean
Dim bRet As Boolean 'boolean return
Dim lRet As Long 'long return
Dim axMenuID As String
Dim lngToolbarDocTypes As Long
'1. capture SW session and cookie to class-wide variable
' store reference to SW session
Set axSldWorks = ThisSW
' store cookie from SW
axCookie = Cookie
2. set Addin Callback info
'Inform SW about the object that contains the callbacks
bRet = axSldWorks.SetAddinCallbackInfo(App.hInstance, Me, axCookie)
'App.hInstance = program's handle
'Me = this class module
3. add menus
axMenuID = "MFs"
 lRet = axSldWorks.AddMenu(swDocASSEMBLY, axMenuID, 6)
 'b. set up menu strings
 'we have two functions, we need two menu picks
Dim axMenu1 As String
axMenu1 = "MFs Form@MFs@" & axMenuID
'c. add menus using axSldWorks.AddMenuItem2
```

```
'this call needs to be made once for each document type:
'NONE, Part, Assembly, and Drawing
'PARTS:
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu1, 0, "CallAssembly", "EnableIfAssembly", "Add Workpiece.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu2, 0, "CallClamps", "EnableIfAssembly", "Add Baseplate."
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu3, 0, "CallClamps", "EnableIfAssembly", "Add Workpiece.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu4, 0, "CallRiser", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu5, 0, "CallWorkpiece", "EnableIfAssembly", "Add V-Block.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu6, 0, "CallVblock", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu7, 0, "CallBaseplateform", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocPART, axCookie, axMenu8, 0, "CallBaseplateform", "EnableIfAssembly", "Add Baseplate.")
'ASSEMBLIES:
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu1, 0, "CallAssembly", "EnableIfAssembly", "Add Workpiece.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu2, 0, "CallClamps", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu3, 0, "CallClamps", "EnableIfAssembly", "Add Workpiece.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu4, 0, "CallRiser", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu5, 0, "CallWorkpiece", "EnableIfAssembly", "Add V-Block.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu6, 0, "CallVblock", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu7, 0, "CallBaseplateform", "EnableIfAssembly", "Add Baseplate.")
bRet = axSldWorks.AddMenuItem2(swDocASSEMBLY, axCookie, axMenu8, 0, "CallBaseplateform", "EnableIfAssembly", "Add Baseplate.")
```

Figure 5-7 shows the MFs menu created in SolidWorks from the previous add-in code. The created menus perform different functions; they can add components to new or existing SolidWorks documents, perform the assembly process for the added components, or open new user interfaces for controlling different SolidWorks functions. After writing the add-in code and adding the proper VB modules and forms to the ActiveX DLL project, a **.dll** file is created and copied to the SolidWorks directory. Then, this **.dll** file is opened in the SolidWorks environment to apply the functions to the developed menus.

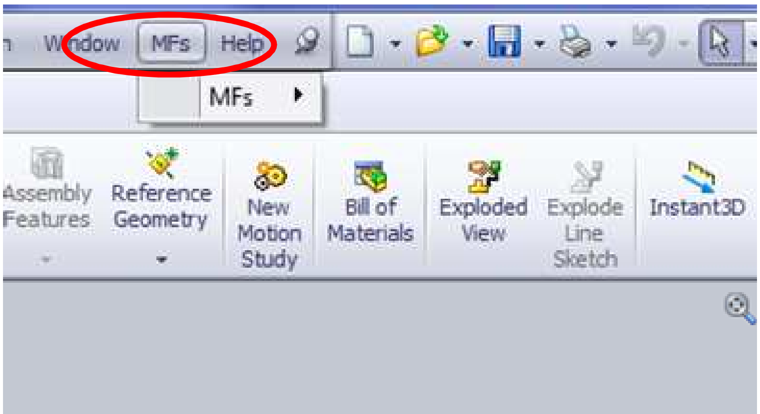


Figure 5-7. An example of the MFs menu developed in SolidWorks.

5.3.1 Assembly Simulation by Macros

In order to automate the fixture elements assembly process in SolidWorks, macro recording functions are used for simulation purposes. Macro recording is an excellent method for automating the design and assembly process in SolidWorks. However, these macros can only be applied to the master SolidWorks document in which they have been recorded. The solution to this problem is to create global macros by modifying the recorded macros. This is completed by changing the VB methods and classes of the recorded macros and adding **swConst** modules and **swAssembly** or **swPart** class modules. This makes the macros available to any SolidWorks document. However, these global macros are still not in a suitable format for the ActiveX DLL project developed. The most important function for the add-in projects is how to make the created menus call the global macros which perform the SolidWorks design and assembly commands. This is achieved by importing the global macros into the ActiveX DLL project as modules in a **.bas** format, then writing a subroutine code for each macro, and finally calling this subroutine by the specific menu's icon. In the previous code, for example, "Call MFs Form" is the subroutine of the macro used for opening "MFs Form" that is called by the menu named "MFs".

5.3.2 Assembly Simulation Implementation

The assembly simulation starts by creating the macros for each fixture element and the assembly steps. The macros for adding the fixture elements are creating first. For more flexibility, a main menu called Modular Fixture System is built. Then two sub-menus are developed called Side Clamping and Top Clamping. The user can select which type of clamping system should be applied by using the knowledge base of the expert system. Both sub-menus are extended to other menus to access information on their related fixture elements for this specific kind of fixturing (**Figure 5-8**). For side clamping the following elements are used:

- Baseplate;
- V-block;
- Pivoting clamp;
- Backstop;
- Riser block;
- Other accessories.

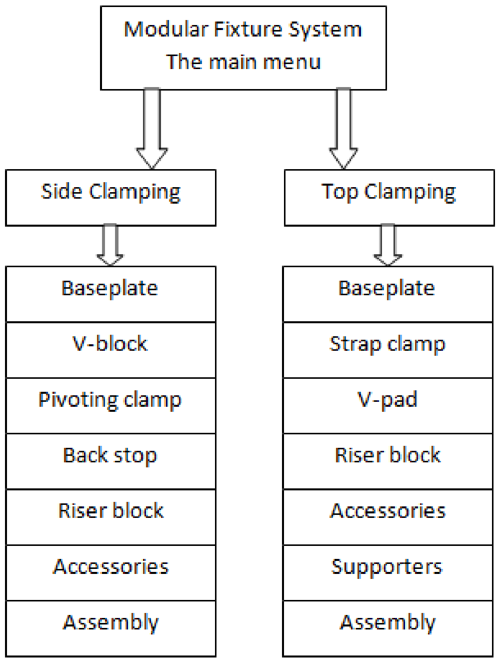


Figure 5-8 . The main and extended menus developed.

For each of these elements a menu should be created. Then, by selecting the specific menu a window is opened for more details. For example, when clicking on the baseplate menu, a window of this element is opened to help the user select the proper baseplate (**Figure 5-9**). The window contains command buttons for available baseplates that could be used within the specified range of the machined area considered in the expert system. Each command button is highlighted by an icon of the specified baseplate, and an ID number is used for choosing the correct baseplate. The interface provides the flexibility for defining the location of the baseplate in the X, Y, and Z directions in the SolidWorks environment. Then the user marks the check box for the selected baseplate and then clicks on the command button.

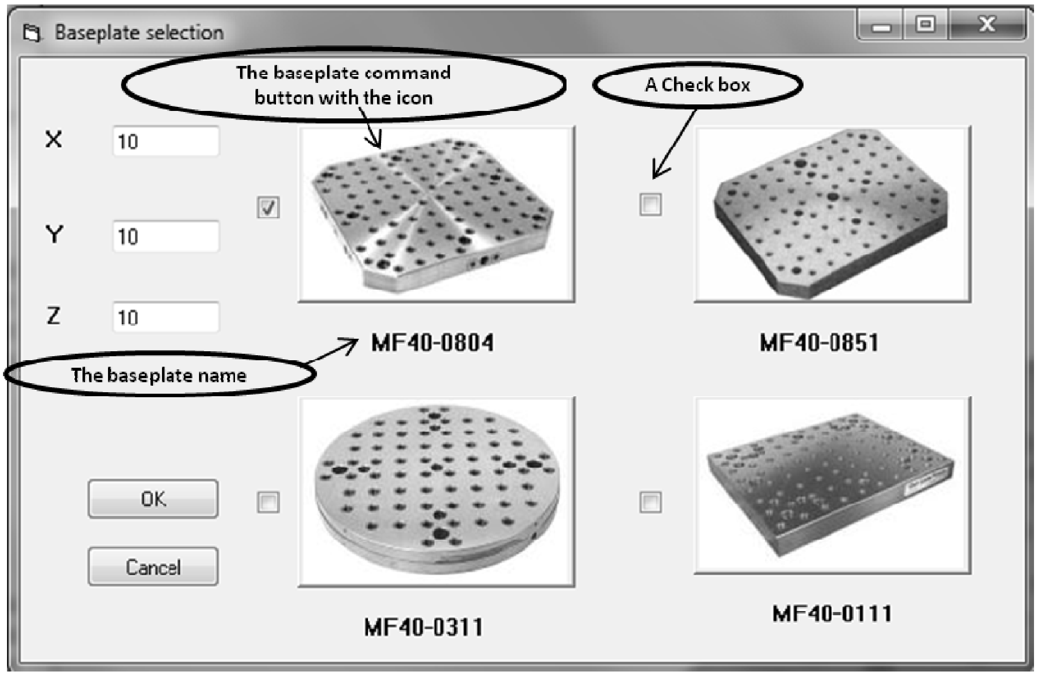


Figure 5-9. The baseplate selection interface. Four kinds of baseplate are included.

After completing the selection and adding the elements to the SolidWorks environment, these elements need to be assembled. A menu called “Assembly” is created and added to the list of the extended menus, as shown in **Figure 5-8**. By clicking this menu, an assembly interface is opened (**Figure 5-10**). This interface contains command buttons for executing the assembly simulation for the fixture elements. In order to activate the command buttons in the baseplate selection interface, a macro is created as shown below:

```
Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swApp =Application.SldWorks
Set Part = swApp.ActiveDoc
boolstatus = Part.AddComponent("C:\Users\Desktop\baseplate -MF40-0804)
End Sub
```

This macro adds baseplate MF40-0804 to the active SolidWorks application document named **swApp**. This macro should now be transferred to the global form by attaching the **swConst** module and **swAssembly** or **swPart** class modules (**Figure 5-11**). The **swConst** module contains all the definitions for the SolidWorks API functions including the properties and methods. The **swAssembly** and **swPart** class modules define to which type of SolidWorks document the specific macro will be applied. The remaining macros for all the fixture elements are provided in Appendix 12.

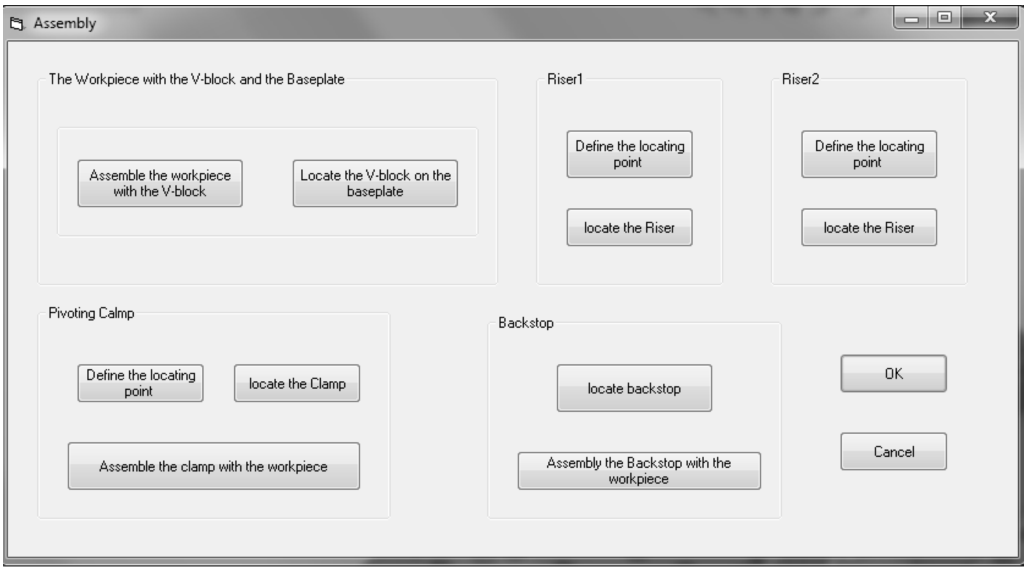


Figure 5-10. The user interface of the fixture element assembly process.

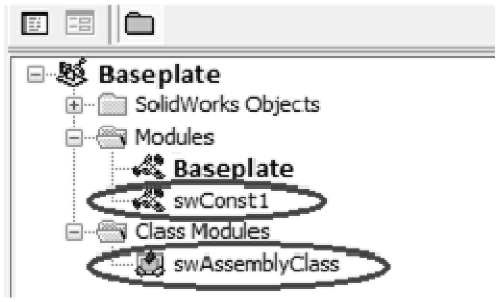


Figure 5-11. Adding the swConst and swAssembly modules to the developed global macro.

The next step is the adding of the macro to the ActiveX DLL project and converting it to the **.bas** code format, which makes the macro available to any SolidWorks assembly or part documents. **Figure 5-12** shows the Baseplate macro in the ActiveX DLL project. The swConst module and swAssembly class modules are also added to this project to improve its availability to any assembly SolidWorks document. The previous process is followed for adding the remaining macros for the fixture elements to the project.

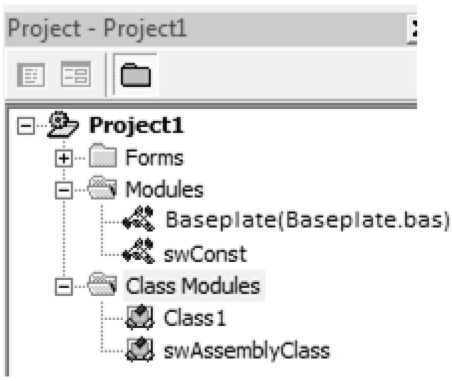


Figure 5-12. The components of the Visual Basic ActiveX DLL project.

Now the simulation of the assembly process begins by taking the V-block and the workpiece as an example. These two elements are added to SolidWorks and then a macro for assembling them is recorded. A tangent mate feature is used between the semi-cylindrical face of the workpiece and the two V-shape faces of the V-block, as shown in **Figure 5-13**. The code for the assembly is:

```
Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swApp = _
Application.SldWorks
Set Part = swApp.ActiveDoc
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1467759532569,
0.2282465512145, 0.2052684525268, True, 1, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.04935040746813,
0.05664959253187, 0.08677583155765, True, 1, Nothing, 0
Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swApp = _
Application.SldWorks
Set Part = swApp.ActiveDoc
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1467759532569,
0.2282465512145, 0.2052684525268, True, 1, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.04935040746813,
0.05664959253187, 0.08677583155765, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = Part.AddMate3(4, 1, False, 0.08962381358722, 0, 0, 0.001,
0.001, 0.5235987755983, 0.5235987755983, 0.5235987755983, False,
longstatus)
Part.ClearSelection2 True
Part.EditRebuild3
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.08743441472234,
0.1639087578332, 0.1920150657936, True, 1, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.146900136872,
0.08290013687201, 0.1404353580714, True, 1, Nothing, 0)
Set myMate = Part.AddMate3(4, 1, False, 0.09399852070358, 0, 0, 0.001,
0.001, 0.5235987755983, 0.5235987755983, 0.5235987755983, False,
longstatus)
Part.ClearSelection2 True
Part.EditRebuild3
End Sub
```

The same procedure is followed for transferring this code to the global form and then to the **.bas** form. **Figure 5-14** shows the macro new_assem11 which is developed and added to the project. The assembly simulation is divided into several steps. The full codes for the fixture element assembly simulation are given in Appendix 13.

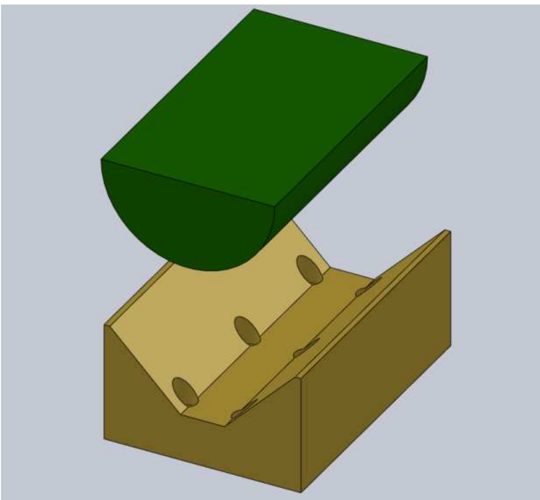


Figure 5-13. The V-block and the workpiece assembly.

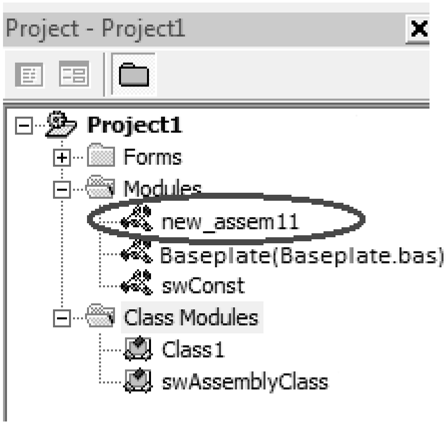


Figure 5-14. The assembly macro new_assem11 in the ActiveX DLL project.

For compiling the command buttons in the user interfaces two approaches are considered. The first approach is to call the macro for each element by using the subroutine:

```
Public Sub cmdBaseplate_Click ()
Baseplate.main
End Sub
```

This subroutine calls the macro baseplate by clicking the command button to add this element to the SolidWorks environment. The second approach is to write a code in order to compile the command button:

```
Public Sub cmdBaseplate_Click ()
Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swApp = Application.SldWorks
Set Part = swApp.ActiveDoc
boolstatus=
Part.AddComponent("C:\Users\Desktop\Baseplate.SLDPRT",
0.1690002083701, 0.2903048910654, -0.02491659965801)
End Sub
```

Both approaches are correct and compatible. To compile these approaches it is important to attach the macros in the **.bas** format to the master directory of the ActiveX DLL project. The menus developed in the SolidWorks environment are shown in **Figure 5-15**.

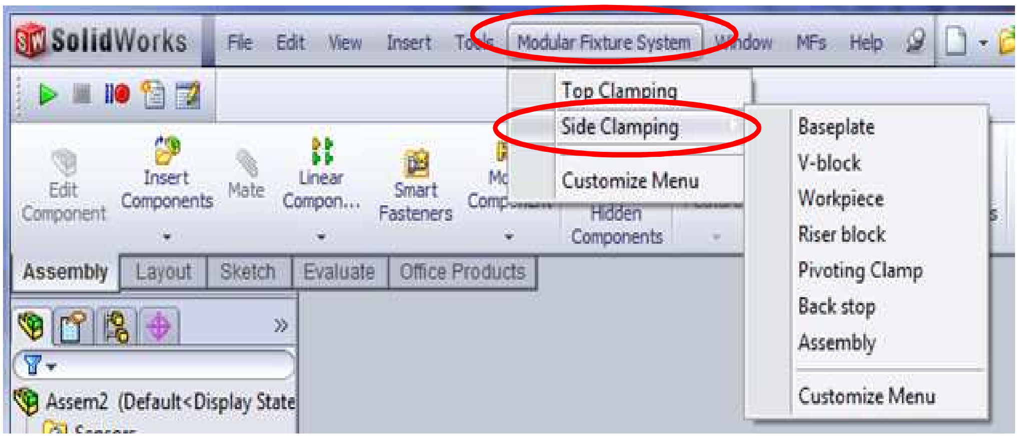


Figure 5-15. The main and extended menus developed for the system.

5.4 Results and Discussion

Simulation by macros is employed for developing the user interface for the related fixture elements. **Figure 5-16** shows the selection of the side clamps and backstops. The interface for the V-blocks selection is shown in **Figure 5-17**, while **Figure5-18** illustrates the riser block interface. To enhance the system's flexibility, other user interfaces are developed in different approaches because there are more details that need to be considered regarding the fixture elements included in these interfaces. From **Figure 5-19**, strap clamps can be selected for top clamping. These clamps are considered subassemblies, as well as the supporters in **Figure5-20**. The accessories necessary to complete the assembly process of the fixture elements are illustrated in **Figure 5-21**. Strap clamps and supporters are used for the top clamping system. This system is not explained in this research as the side clamping system was the base of the MF system developed. The main components for side clamping are the baseplate, pivoting clamp, backstop, riser block and V-block.

Figure 5-16. The selection window for the side clamps and backstops.

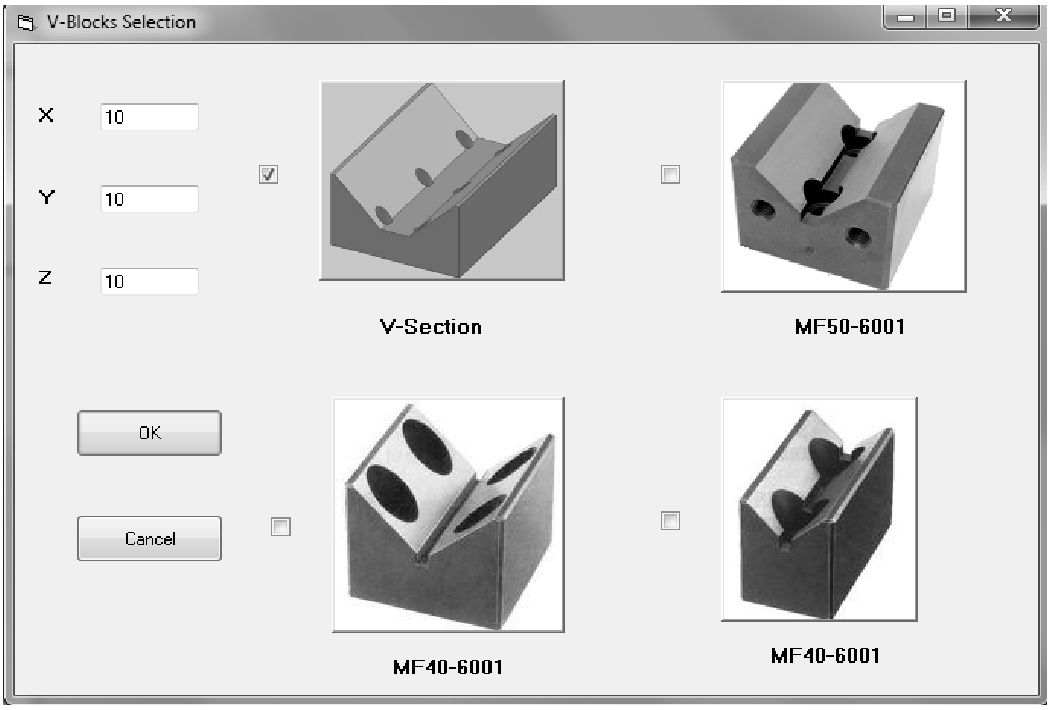


Figure 5-17. The selection window for the V-blocks.

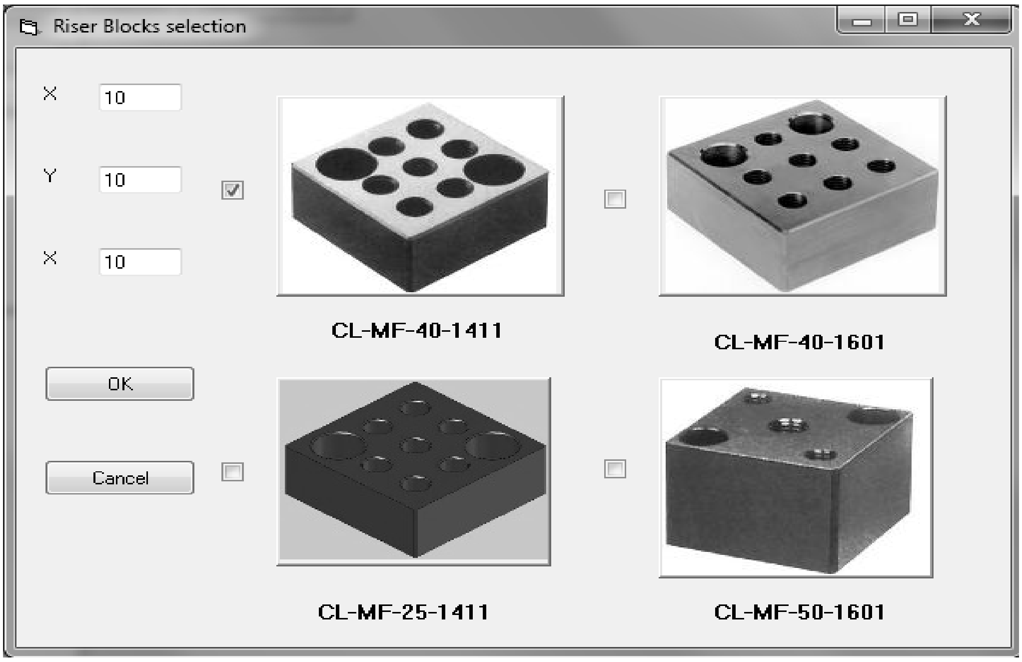


Figure5-18. The selection window for the riser blocks.

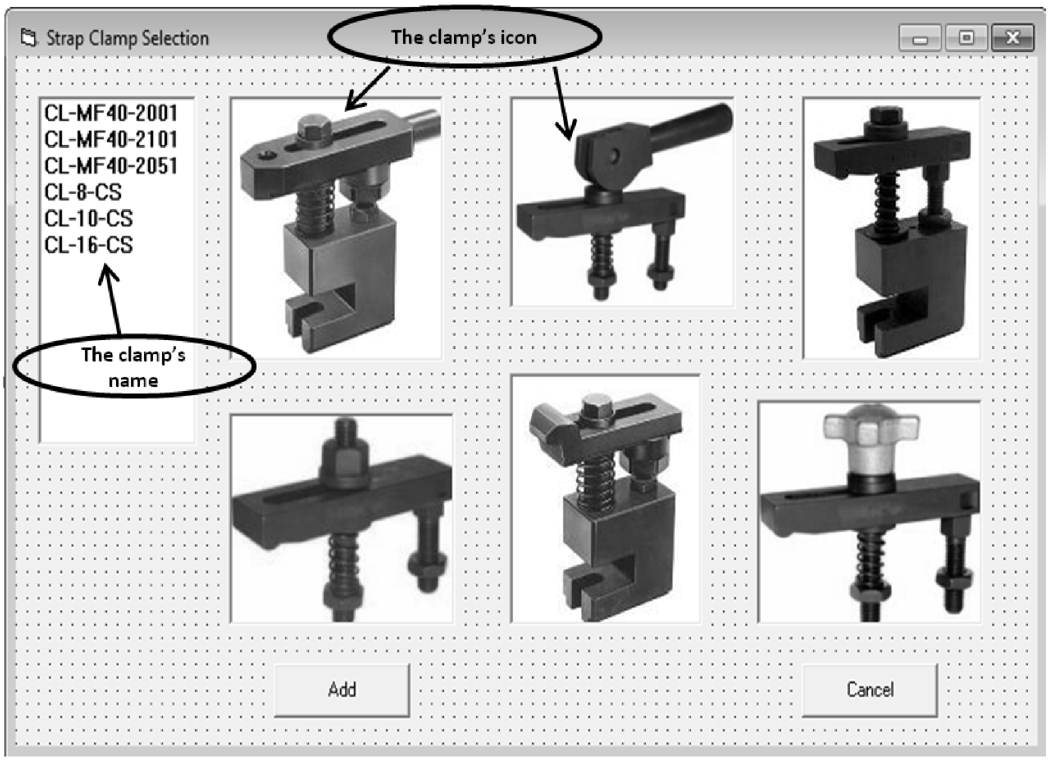


Figure 5-19. The selection window for the clamps for the top clamping system.

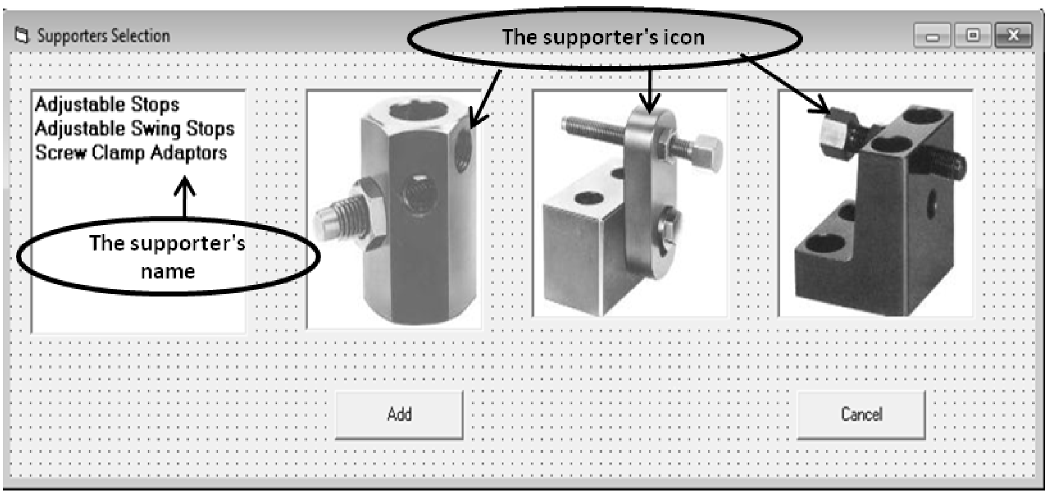


Figure5-20. The selection window for the supporting adapters.

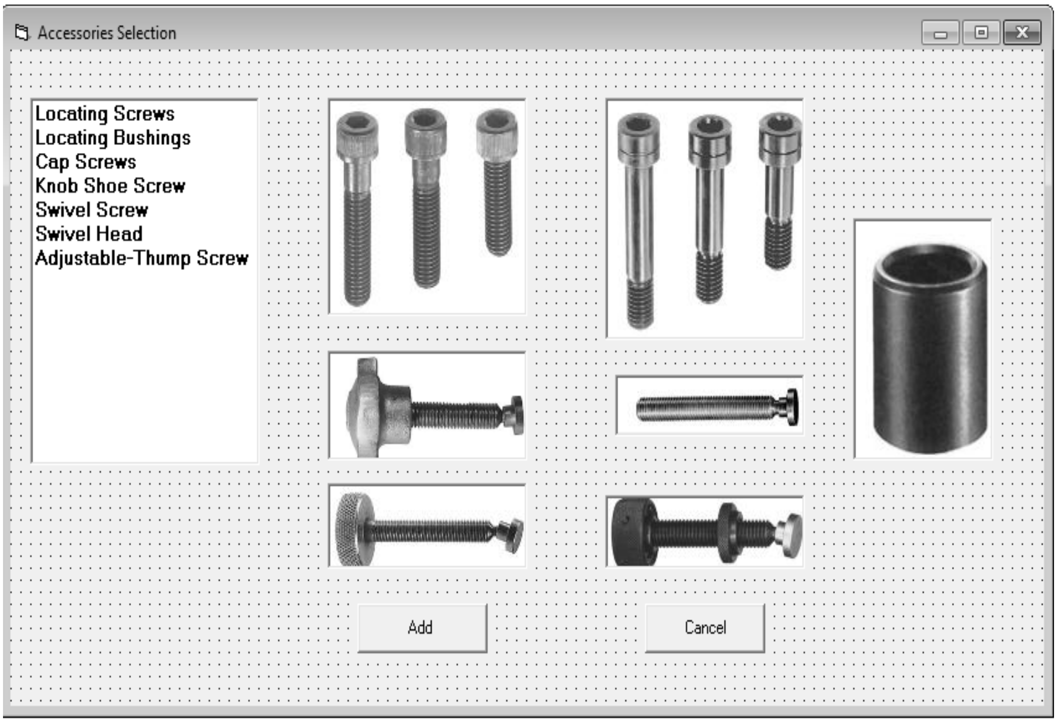


Figure 5-21. The selection window for the accessories to complete the fixturing system.

The assembly process starts after adding the components into the SolidWorks environment, as shown in **Figure 5-22**. Then, macros are developed for appropriately assembling the two parts as explained previously for the workpiece and the V-block. The macros created for the system are used in order to assemble:

- The workpiece with the V-block.
- The V-block with the baseplate.
- The riser block with the baseplate (two macros).
- The pivoting clamp with the riser block.
- The pivoting clamp with the workpiece.
- The backstop with the riser block.
- The backstop with the workpiece.

Therefore eight steps are taken to complete the assembly simulation. The process is performed in this way because errors could occur if the whole process was undertaken in one step. This is related to problems in SolidWorks when applying rotating and reviewing functions on components during the assembly of mating features. The positions and directions of the elements are predefined in the VB codes. For more effective development, these positions and directions could be controlled by creating extra codes. Due to time limitations, this function was not considered here but could be included in future work. By completing the eight assembly steps the final MF layout determined, as shown in **Figure 5-23**. After finishing the assembly, interference detection is executed to ensure that there are no interfaces between the fixture components. This process is completed by SolidWorks as shown in **Figure 5-24**. If any interference is found, the mating features such as choosing the correct edges or faces should be revised or modified to achieve an efficient assembly.

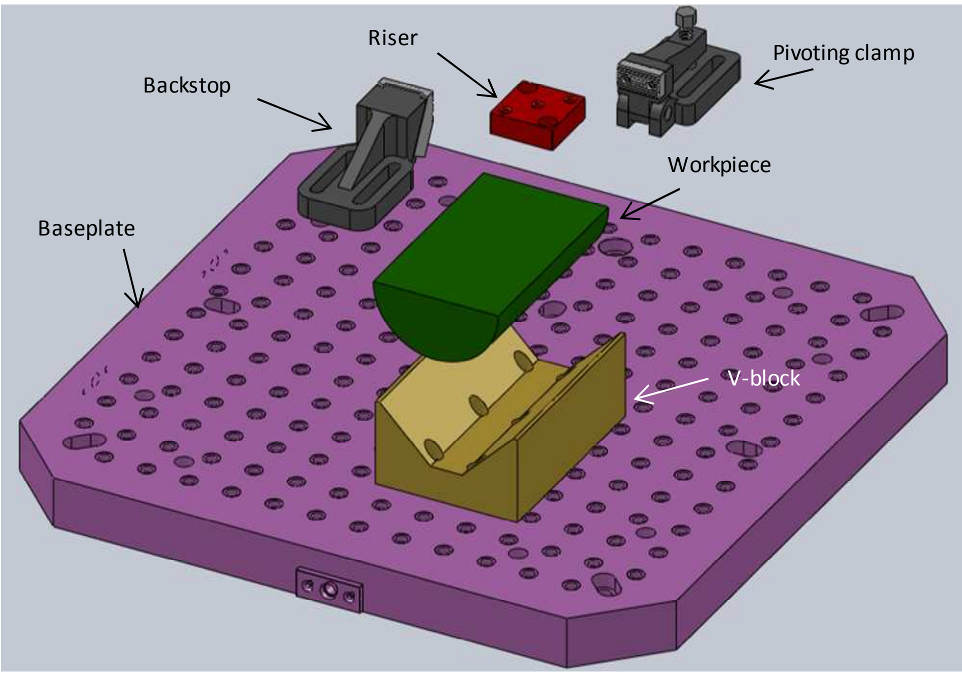


Figure 5-22. Modular fixture elements added in SolidWorks modelling environment from the menus.

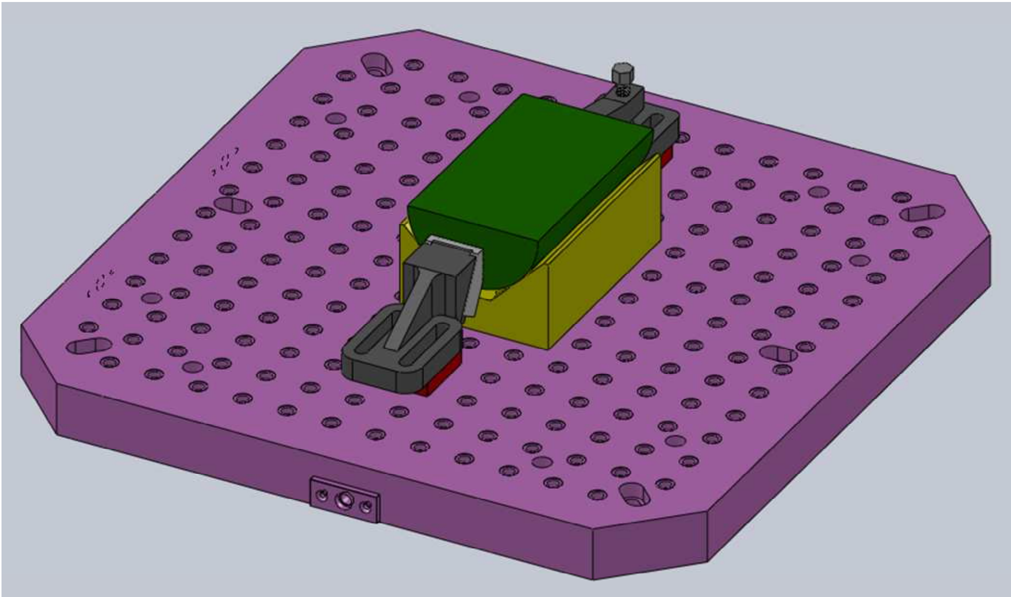


Figure 5-23. The complete layout of modular fixture elements generated after assembly.

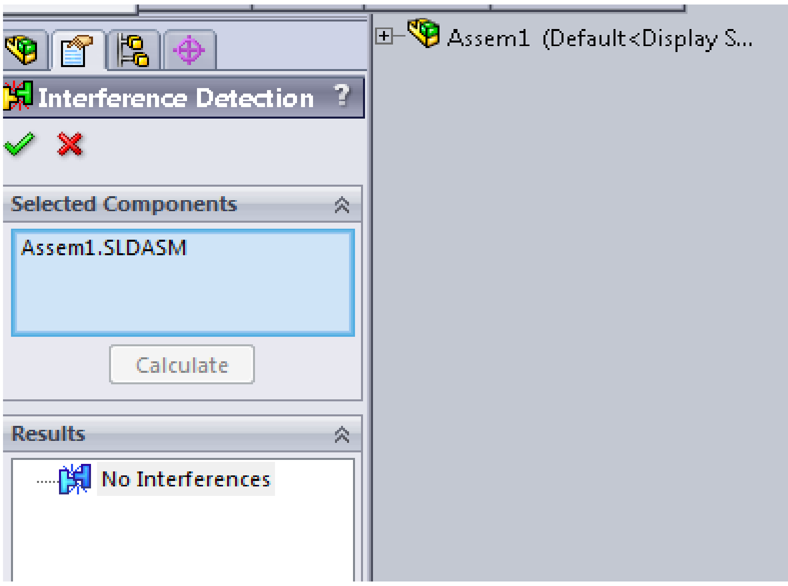


Figure 5-24. The interference detection tool applied in SolidWorks.

5.5 Summary

The assembly process for MF design is explained for a specific workpiece. The knowledge base is developed for this process and is related to side clamping modular fixturing. Moreover, SolidWorks API is employed for automating the assembly process in SolidWorks. The simulation of this process is completed by using macros functions in SolidWorks. For this purpose, an ActiveX DLL VB project is created for developing new menus in the SolidWorks environment. From these menus, user interfaces are expanded so that they can be opened by clicking each of the menus, thus allowing the user to select and insert the correct elements. The approach of applying SolidWorks API is explained in detail, and the system is tested by using the interference detection tool included in SolidWorks to be sure that no interfaces have occurred between the MF elements.

Chapter Six

6. SolidWorks Knowledge Base and the Features Library

This Chapter presents the concept of using the DriveWorksXpress tool in SolidWorks for design and assembly automation purposes. This tool is applied to generate new models for the existing fixture elements, and new assembly configurations can be also generated by using this tool. In addition, the process of constructing the feature library is explained. The feature library of modular fixture elements is an important component. This component is necessary to make computer-assisted modular fixture design more automated. The fixture elements feature library can be constructed by using different methods. Some of these methods use interfacing software such as C++ with CAD software. However, some CAD software provides an opportunity to directly develop a feature library; SolidWorks is one such software which is provided with capabilities to construct feature library. The feature library in Solidworks includes standard fixture elements that can be directly downloaded from websites. In addition, it can also include new fixture elements which are designed through SolidWorks. Finally, an approach for estimating the cutting and the reaction forces is developed under maximum clamping forces. Based on this estimation, a finite element simulation is conducted for secure placement of the fixture elements.

6.1 DriveWorksXpress

DriveWorksXpress is a tool that is implemented in SolidWorks. The advantages of using this tool are the ability to:

1- Automatically create a feature library of the elements or models;
2- develop new models by modifying the dimensions of exciting elements or models; and
3- control different configurations based on the rules.

Based on which DriveWorksXpress is considered a tool to automate the design process in SolidWorks [60]. This approach was applied for building the knowledge base for the grid assembly project by implementing DriveWorksXpress 6 with SolidWorks 2007. A user interface was developed that asked the user to specify the design parameters based on the rules in order to generate new models [61]. This tool is applied in this research in order to investigate the capability of SolidWorks in terms of automating the MF design and assembly. To implement DriveWorksXpress for this purpose, the following steps are followed:

Step 1: A new SolidWorks window is opened.

Step 2: From the Menu bar and then Tools, DriveWorksXpress is chosen. A new window appears for this tool with five tabs, **Welcome, Capture, Form, Rules** and **Run** (**Figure 6-1**).

Step 3: In the welcome window there are three choices, **Create/Change Database, Add/Edit Models** and **Run Models**. For the first use of DriveWorksXpress, Create/Change Database is chosen and then **Next**.

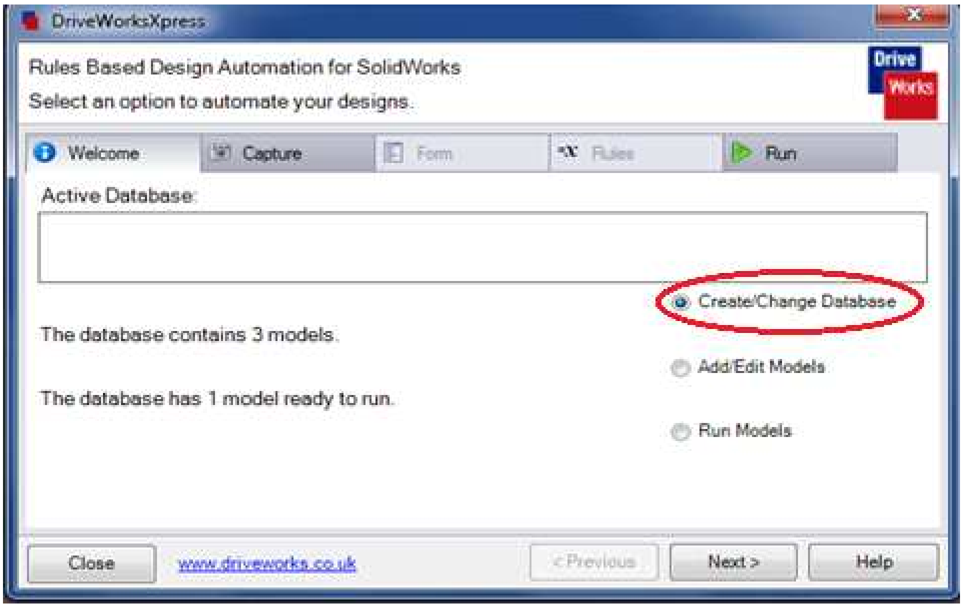


Figure 6-1. The welcome window of DriveWorksXpress.

Step 4: The name and location of the feature library are specified in the new window in **.mdb** file format and then **Open** is selected**.**

Step 5: After that, from the capture tab, **Browse for new model** is chosen (**Figure 6-2**) and SolidWorks part or assembly files are selected. This results in capturing all the components in these files (**Figure 6-3**) and the feature library is automatically created for these components. The feature library is saved to the path already specified in step 4.

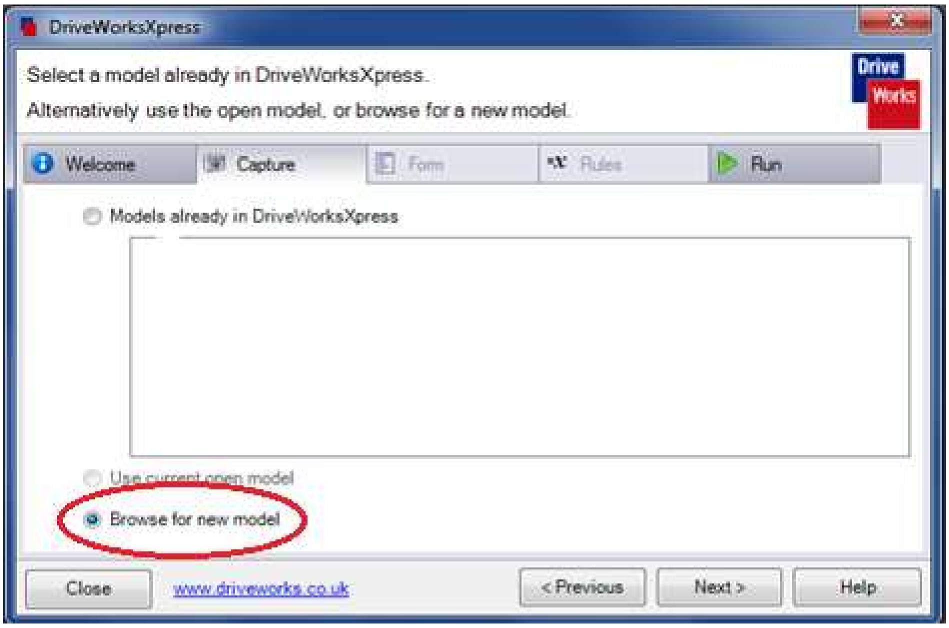


Figure 6-2. The capture window in DriveWorksXpress.

Establishing the feature library using DriveWorksXpress tool creates the necessary tables of the components. The tables consist of the necessary information about these components and their related relationships, including the input and output data (**Figure 6-4**). The tables can be modified and new components and input data can be added. The setting of the feature library can be changed by applying Microsoft Office Access features.

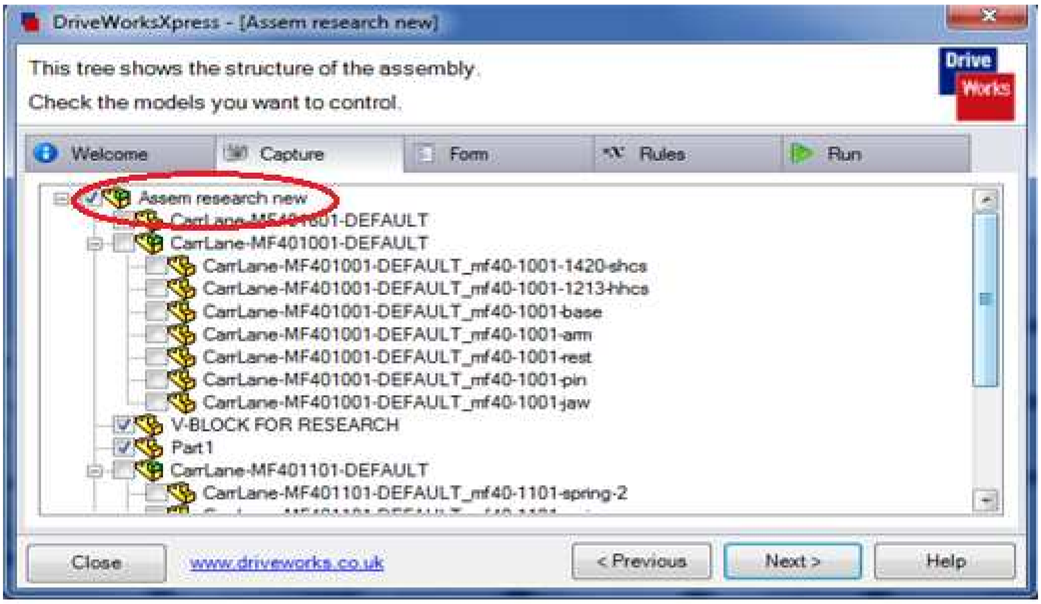


Figure 6-3.The captured assembly model.

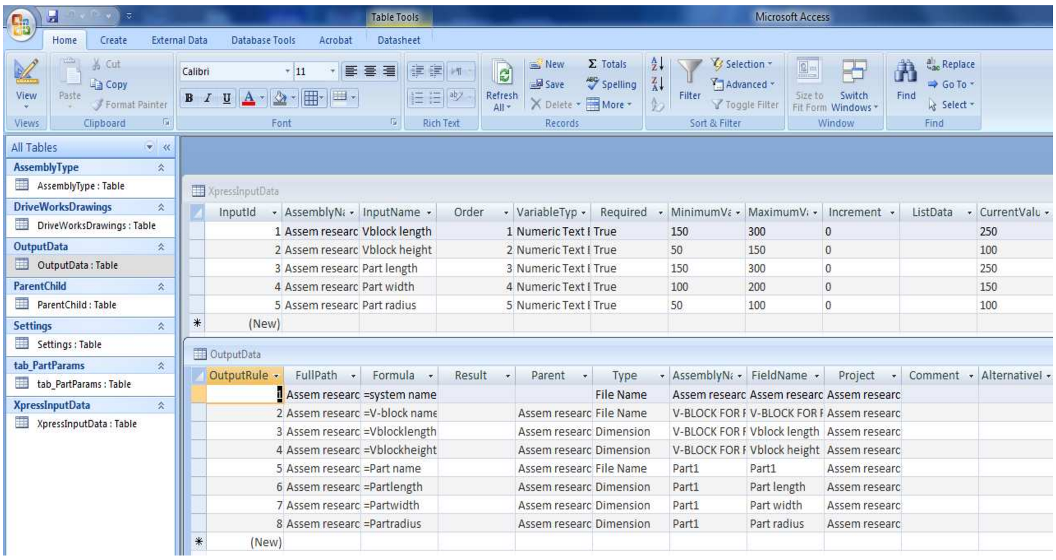


Figure 6-4. The feature library developed in DriveWorksXpress.

Step 6: Now it is important to specify which elements in the assembly need to capture their specifications, dimensions and features in order to develop new elements. It is also possible to capture the properties of the models or the elements and create new drawings. Moreover, in order to control the configurations, the “Yes” option is selected in the configuration tab (**Figure 6-5**).

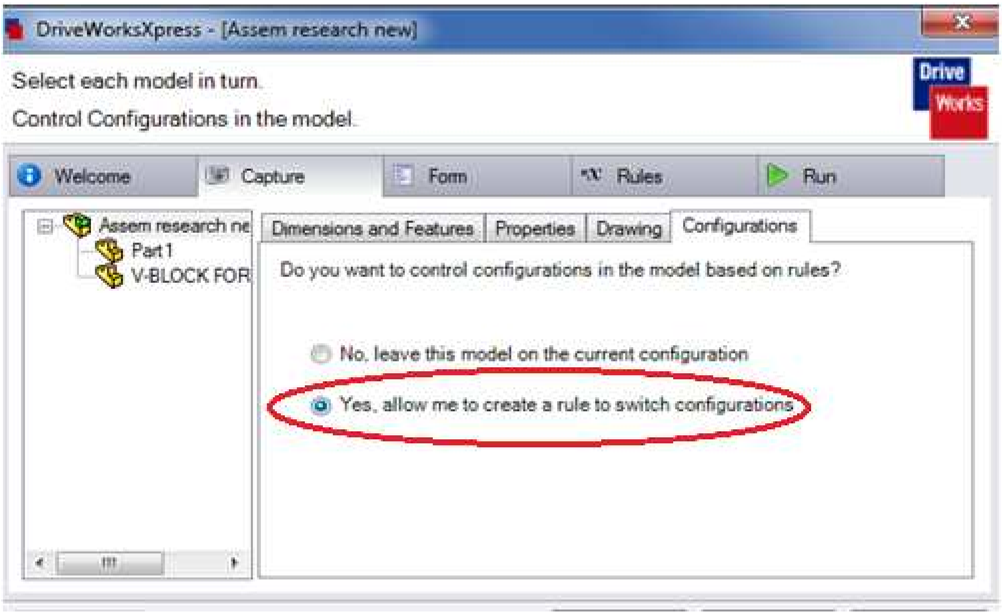


Figure 6-5. Controlling different configurations in the model based on rules.

Step 7: A form is built for the requirements that are needed for creating new elements or models. Five types of forms can be selected; Text Box, Numeric Text Box, Dropdown, Spinbutton, and Check Box. Several kinds of form can be selected at the same time, and new forms can be added and existing forms can also be edited (**Figure 6-6**).

Step 8: The rules are built in order to control the selected requirements as listed below:

- File name
- Configurations
- Custom properties
- Dimensions
- Features

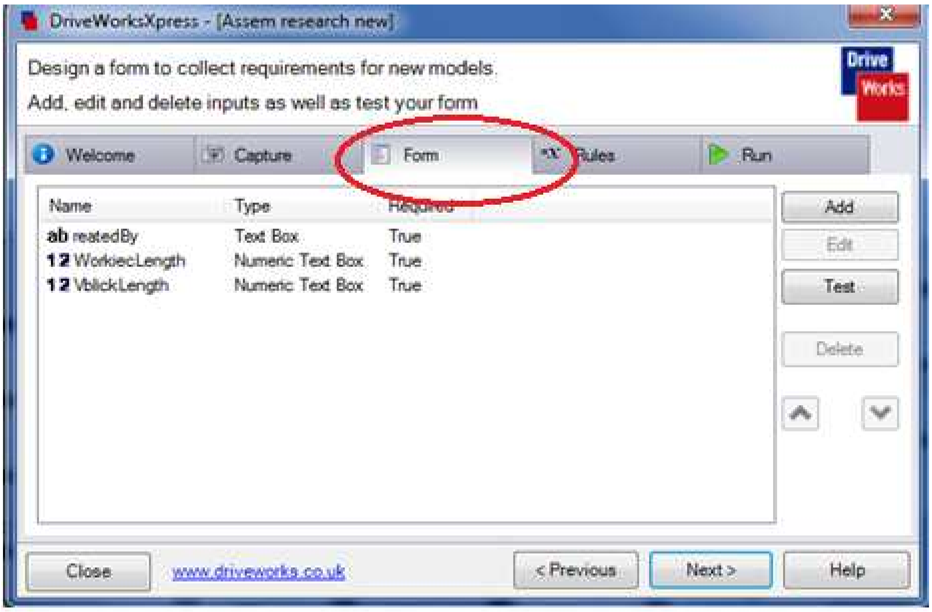


Figure 6-6. Building a form to gather the requirements for the new model.

Some rules are built automatically by DriveWorksXpress, while the missing rules need to be completed. From the Rule tab, each kind of rule can be selected and edited separately or all the rules can be edited at the same time. To edit all the rules, a tree or a list of these rules is shown and each rule is selected to open the rules' editor window (**Figure 6-7**). From this widow, the rules can be created based on:

- Previous input
- Recent rules
- Mathematical relationships
- Logic relationships

Creating rules from the inputs is suitable for modifying the dimensions of the model. In this research, these rules are used for developing new models of the chosen workpiece. Creating rules from mathematical relationships is based on mathematical formulas such as +, -, /, and x, while rules from logic are based on comparison operators such as <, >, <>, =, =<, =>, and the IF-THEN format. This kind of rule is convenient for controlling the configurations of the models and the customised properties of the models or elements.

Step 9: After completing the rules, a new version of the models or elements is generated by running the forms developed (**Figure 6-8**), by specifying the required values for each form. Then, the new models are saved as SolidWorks part or assembly files.

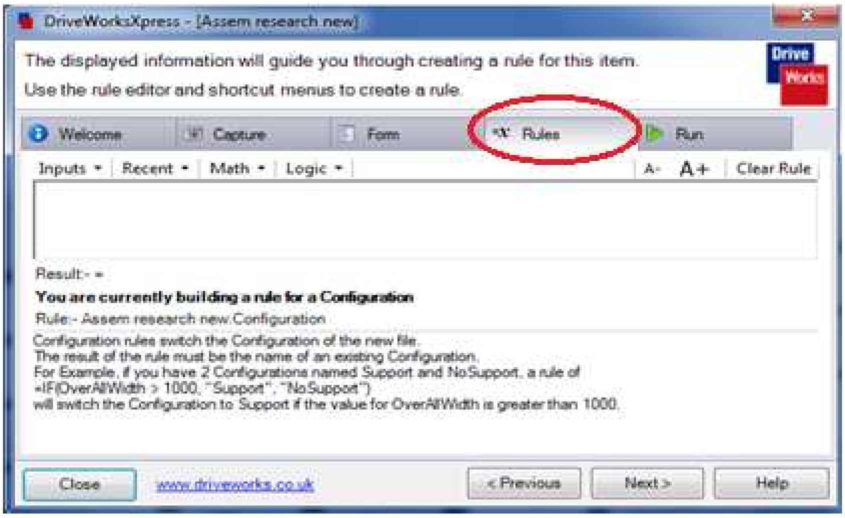


Figure 6-7. The rules editor window. Information is displayed for rules guiding explanation.

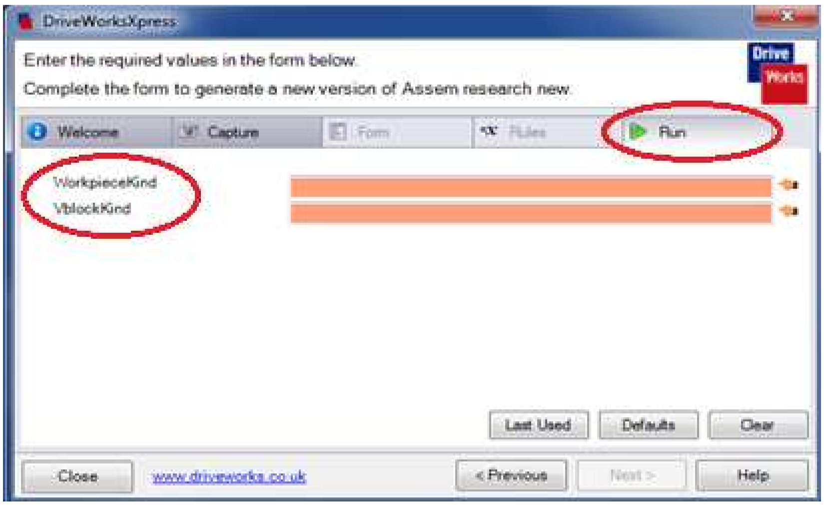


Figure 6-8. Generating a new model by entering the required values.

6.2 SolidWorks Feature library

The feature library of MF standard elements is constructed in SolidWorks. This is completed with the use of the design library features in SolidWorks, which provides the flexibility of storing, managing, and selecting the fixture elements in order to simplify the design and assembly process. The design library procedure is as following:

Step 1: The specific SolidWorks file for the fixture element is opened, and from the right side of the SolidWorks window the **Design Library** is chosen (**Figure 6-9**).

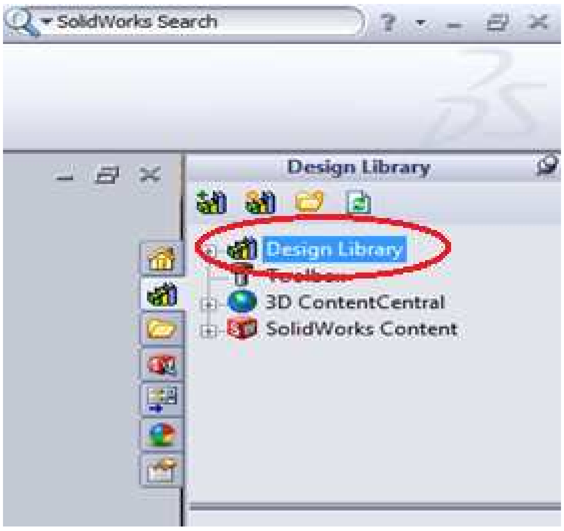


Figure 6-9.The task pane in the SolidWorks window.

Step 2: The design library icon is extended to show all the default folders in the SolidWorks library (**Figure 6-10**). After that, by right clicking on the design library icon, a **New Folder** is chosen. A new folder is created; this folder is renamed **Modular Fixtures**.

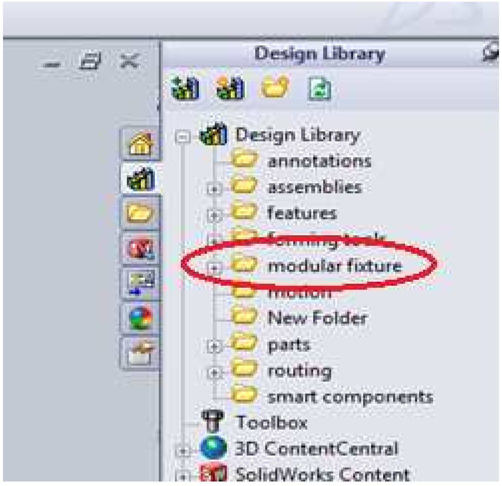


Figure 6-10. Creating the Modular Fixtures folder.

Step 3: By right clicking on the Modular Fixtures folder, **New Folder** is again chosen. This new folder is renamed depending on the fixture element type. For example, if the fixture element is clamp, the folder name will be **Clamps**. The same procedure is followed for the other elements. This helps classify the feature library and gives fast access to the elements (**Figure 6-11**).

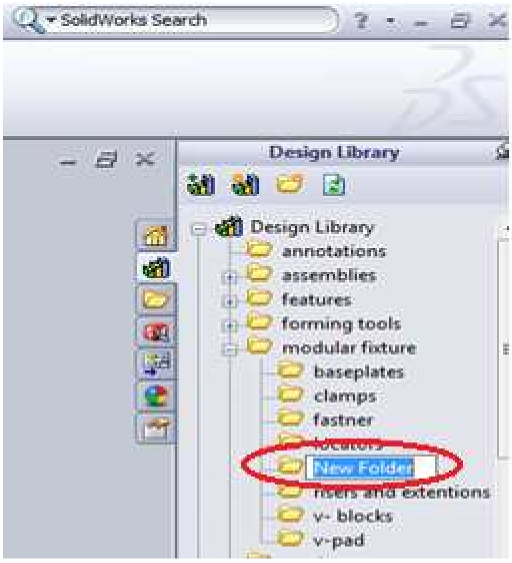


Figure 6-11. Creating a new folder for a fixture element in the Modular Fixtures folder.

Step 4: By clicking on the **Add to Library** icon ( ) in the task pane, the **Add to Library** property manager is opened. The fixture element is selected in **Items to Add**, then the folder in which the element is to be included in the feature library is selected; if the element is clamp, it should be in the clamps folder. The file type can be changed and a description can be included (**Figure 6-12**).

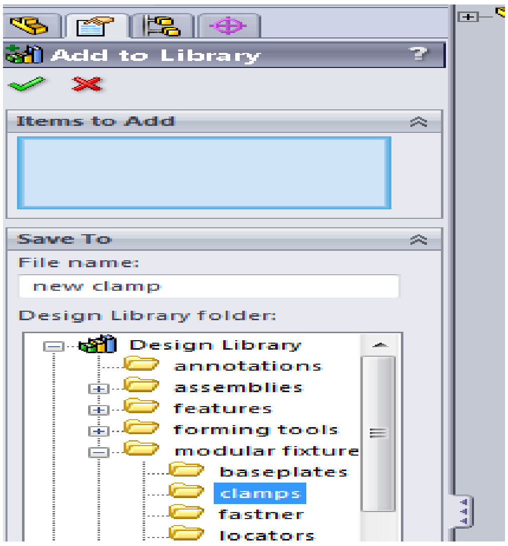
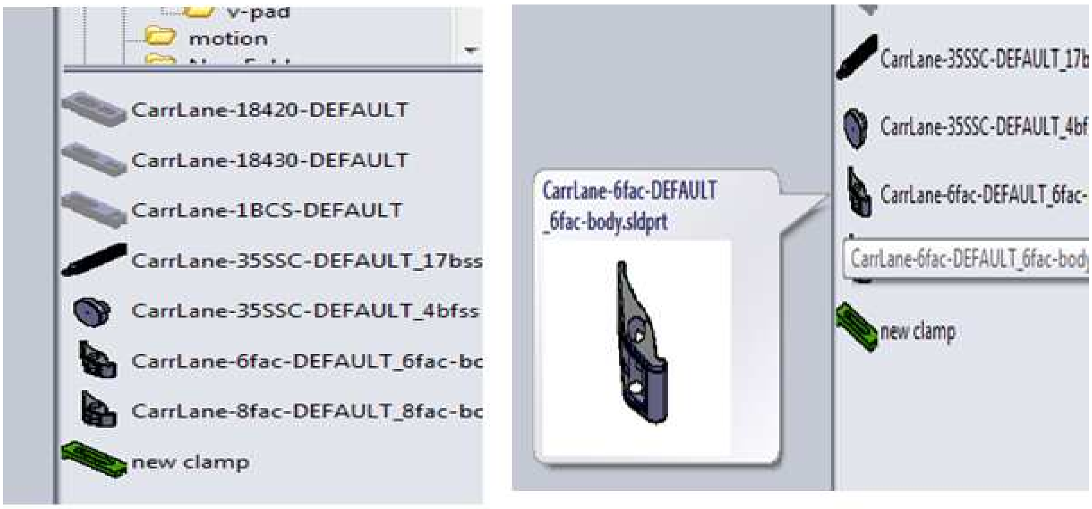


Figure 6-12. Add to Library property manager window.

Step 5: The fixture element is now added to the Modular Fixtures feature library. To re-use this element again in the same modular fixture configuration or in different configurations, the design library is again chosen and then the Modular Fixtures folder. From the specific folder (such as clamps) the element is selected from the list that appears in the task pane, and then dragged into the SolidWorks environment (**Figure 6-13** (a)). If an assembly file is opened, the element will be inserted directly into the window, while if a part file is opened, a new window for the element will be opened as a separate part file. In order to make the feature library more efficient, it is important to classify the specific folders of the feature library. For example, the clamps folder in the Modular Fixtures folder is classified into folders for top clamps and side clamps. Another classification is carried out according to the locating and clamping methods and by considering all the related elements under these classifications. The function of the fixture elements is another factor considered when dividing the feature library. Before choosing the element and dragging it from the list to the SolidWorks environment, it is possible to show a small window for the element by aiming the pointer without clicking on the specific element (**Figure 6-13**(b)).This helps the user choose the right element from a variety of similar elements.



(a) (b)

Figure 6-13. a) A list of fixture elements, b) A small window for a specific element.

6.3 Implementation of DriveWorksXpress

As the main scope of this study is the automation of semi-circular parts, additional case studies are investigated by implementing the concept of DriveWorksXpess. One of these case studies includes generating a fixturing layout for a prismatic part as shown in **Figure 6-14**. The idea of this configuration is to facilities the design of the fixture elements with quick change subplates for CNC machines for several manufacturing processes (**Figure 6-15**) using the appropriate clamps.

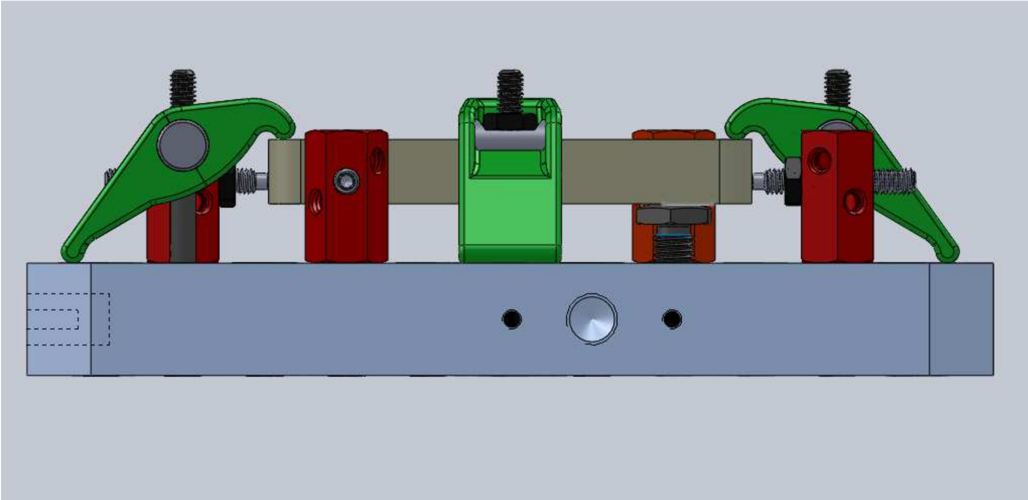


Figure 6-14. Fixturing configuration for the prismatic part (side view).

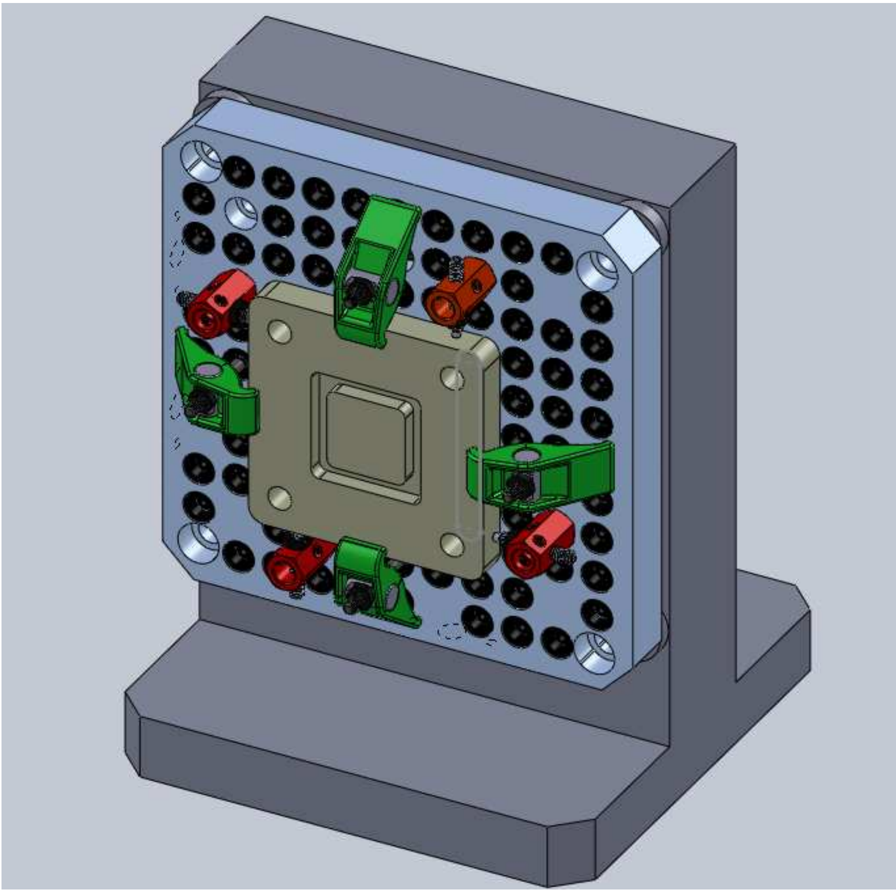


Figure 6-15. Fixturing layout with Quick change Subplate for CNC machining.

The second case study is shown in **Figure 6-16**. This configuration is used to establish the fixture elements layout for irregular parts with the use of strap clamps and the proper supporters.

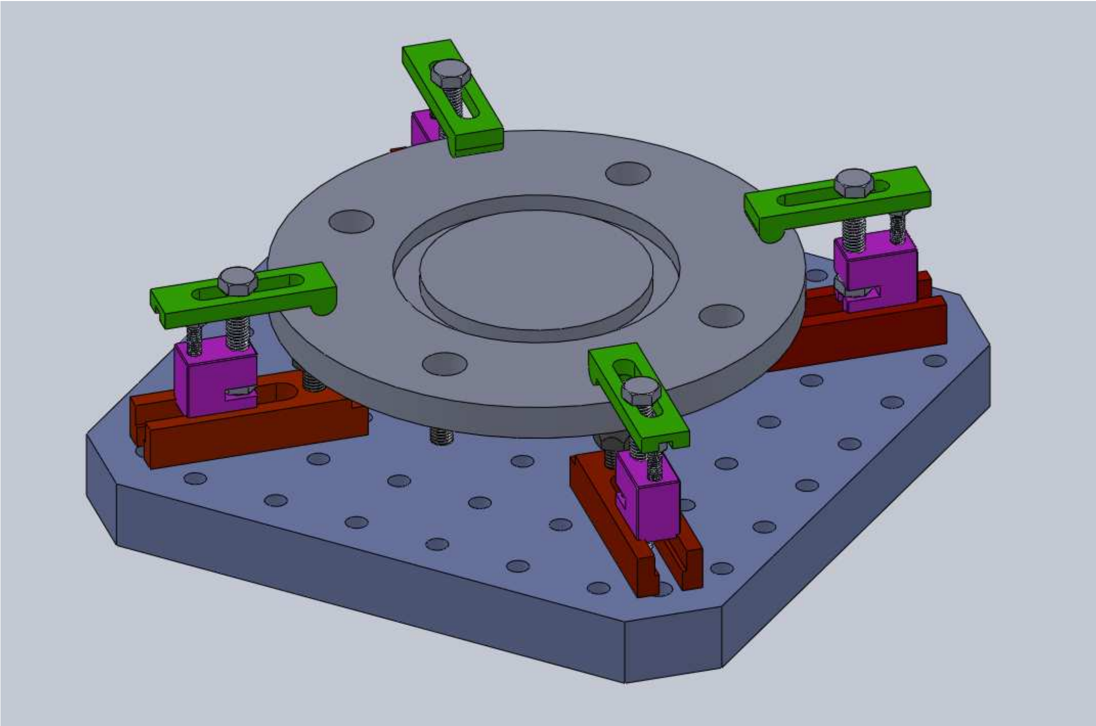


Figure 6-16. Fixturing configuration for irregular part.

6.4 System Stability

It is important to ensure that the fixturing layout is stable and meets the requirements for design and manufacturing purposes. Interference detection already is explained in Section 5.4 between the fixture elements. For safe machining operations and system stability, cutting forces estimation and finite element simulation are also carried out.

6.4.1 Cutting Forces Estimation

Cutting forces are one of the parameters that should be taken in consideration in MF design. Different approaches were applied regarding cutting forces calculation. Isakov described the dynamics of milling by using formulas for cutting force, torque and machining power and introduced "Tangential cutting force theory" based on the dynamics of milling analysis [62]. Other formulas were introduced for calculating tangential, radial and feed cutting forces for milling operations [63]. The calculation approach for cutting forces in this study is based on calculating the tangential force (F_T). The other two forces are estimated from the following formulas based on the F_T value:

The feed force: $F_F = K_F \tan \lambda_s F_T$ [63]

The radial force: $F_R = K_R F_T$ [63]

Where λ_s is the helix angle of the cutter, K_F is the feed force coefficient, and K_R is the radial force coefficients [63, 64].

Form these three forces the resultant force can be calculated as follows:

$$F = \sqrt{F_T^{\ 2} + F_F^{\ 2} + F_R^{\ 2}}$$

Figure 6-17 shows the three forces for the end milling process. By considering F_T, an approach was illustrated by Carrlane based on a two-dimensional static mechanics concept [38].

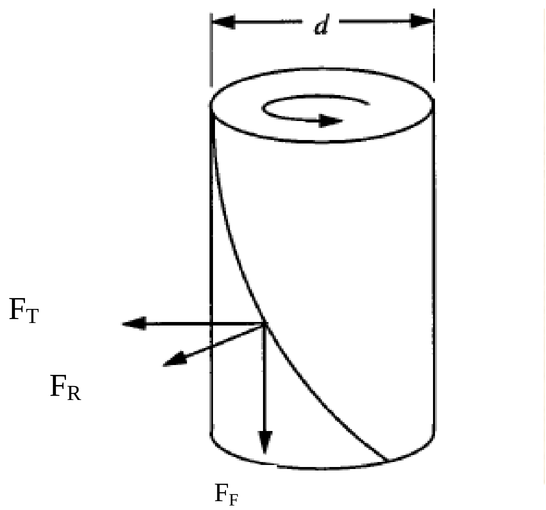


Figure 6-17. The three cutting forces for end milling [63].

This approach was used to estimate the clamping force from the following formula:

$$\text{Clamping force} = \frac{\text{cutting force}}{\text{static friction coefficient}} \text{ x safty factor} \qquad [38]$$

In this study, the maximum clamping forces are known from the specification of the pivoting clamps (Appendix 11). Therefore, a 3D body diagram is considered for the semi-circular workpiece for estimating the maximum tangential force F_T value. **Figure 6-18** shows the 3D force analysis approach for the selected workpiece.

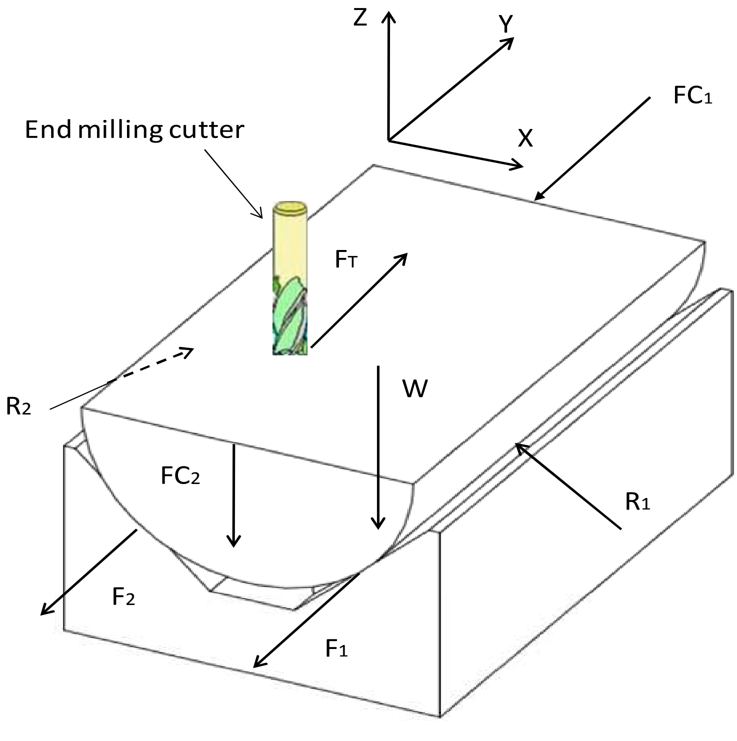


Figure 6-18. The 3D body diagram for the semi-circular workpiece. F_T is in the positive Y direction.

FC_1 is the horizontal clamping force while FC_2 is the vertical clamping force acting on the workpiece. The values of these clamping forces are 8900 N and 980 N, respectively. The workpiece's weight (W) is 15.9 Kg = 156 N (Appendix 8).

R_1 and R_2 are the reaction forces, while F_1 and F_2 are the friction forces on the contact surfaces between the V-block and the workpiece. The pivoting clamp applies a horizontal force FC_1 on one side with two vertical forces FC_2 on both sides. However, the direction of the cutting force F_T is considered to be in opposition to FC_1, therefore the other $FC_2$is ignored on the right side as there is no chance that the workpiece can lift. Now, by applying the three equations for the sum of forces in the three directions (X, Y and Z):

$$\sum F_Y = 0: - F_1 - F_2 + F_T - FC_1 = 0 \quad (1)$$

$$\sum F_Z = 0: - W - FC_2 + R_1 \text{Sin } 45 + R_2 \text{Sin } 45 = 0 \quad (2)$$

$$\sum F_X = 0: - R_1 \text{ Cos } 45 + R_2 \text{ Cos } 45 = 0 \quad (3)$$

Where FC_1 = 8900 N, FC_2 = 980 N, W = 156 N, $F_1 = R_1\,\mu$, and $F_2 = R_1\,\mu$, as μ is the friction coefficient and is equal to 0.3 (**Table 1**), and F_T = tangential cutting force. The friction coefficient is considered for cast iron on cast iron contact surfaces as both the workpiece and the V-block are made from the grey cast iron. The 45°angle is the V angle for the V-block as shown in **Figure 6-19**.

Contact of surface	**Friction coefficient (Dry)**	**Friction coefficient (Lubricated)**
Steel on steel	0.15	0.12
Steel on cast iron	0.19	0.1
Cast iron on cast iron	0.3	0.19

Table 1. Friction coefficients for steel and cast iron [38].

By solving the above three equations:

From equ.2 we get: $0.7\ R_1 + 0.7\ R_2 = 1136$

From equ.3 we get: $-\ 0.7\ R_1 + 0.7\ R_2 = 0$

Therefore $1.4\ R_2 = 1136$, and thus $R_2 = 811.5$ N

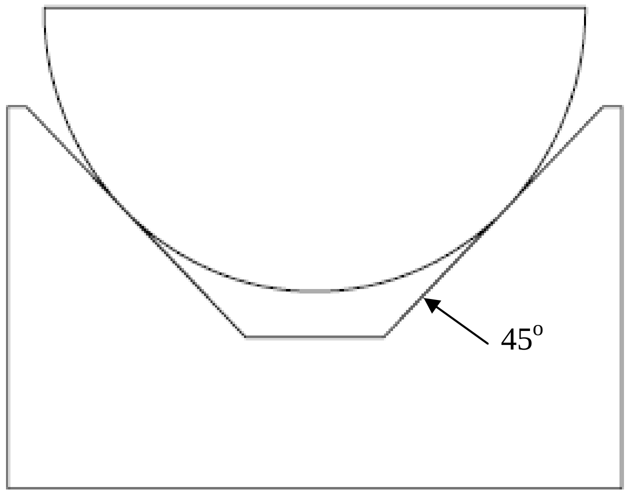


Figure 6-19. The contact surfaces between the workpiece and the V-block.

By substituting in equ.2:

$R_1 = 811.5$ N, therefore $F_1 = F_2 = 811.5 \times .3 = 243.5$ N

By substituting F_1 and F_2 in equ.1 we get: $F_T = 9387$ N

This is the maximum value for F_T in the positive Y direction that can be applied under the maximum clamping forces for this case. As the direction of the cutting forces changes, the same approach is applied by considering F_T in the positive X direction, as shown in **Figure 6-20**.

From this figure, the clamping force FC_1 is eliminated because it does not act against the cutting force. In addition, the reaction force R_2 and the friction force

F_2 are also eliminated because F_T is not acting in their direction. The direction of F_1 is parallel to the contact surface between the workpiece and the V-block and perpendicular to the reaction force R_1. Now, by applying the same equations for the sum of forces:

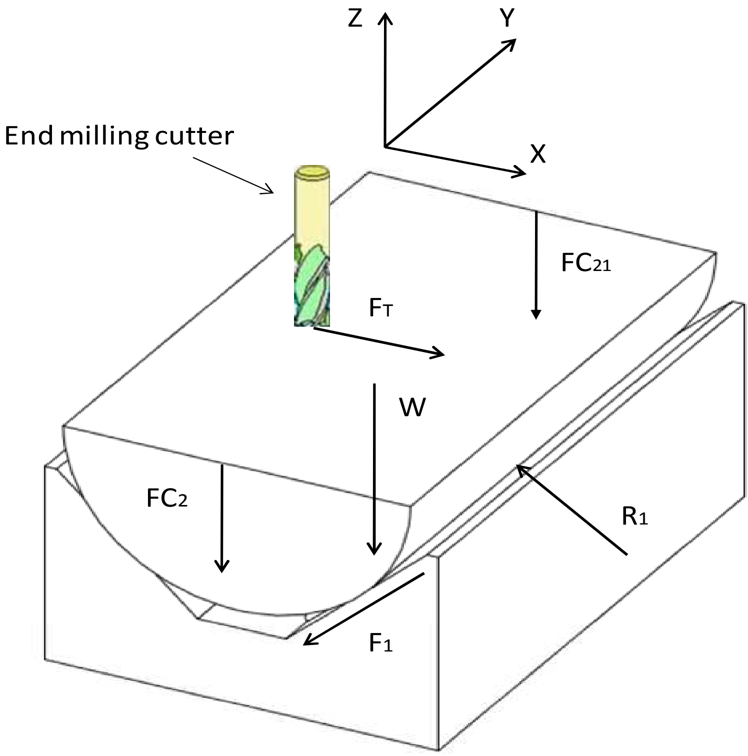


Figure 6-20. The 3D body diagram for the semi-circular workpiece. F_T is in the positive Y direction.

$$\sum F_Z = 0: \ - W - FC_{21} - FC_2 + R_1 \text{ Sin } 45 - F_1 \text{Sin } 45 = 0 \qquad (4)$$

$$\sum F_X = 0: \ F_T - R_1 \text{ Cos } 45 - F_1 \text{Cos } 45 = 0 \qquad (5)$$

$F_1 = R_1\ \mu$, therefore from equ.4 we get: $- 2116 + 0.49\ R_1 = 0$, Then $R_1 = 4318.4$ N, and by substituting in equ.5 we get:

$F_T - 0.91R_1 = 0$, then: $F_T = 3929.8$ N

6.4.2 Cutting and Clamping Forces Simulation

The finite element simulation for this system is completed by the SimulationXpress tool in SolidWorks. This simulation is applied on the workpiece under maximum clamping forces to ensure that there is no deformation that could occur within the workpiece. **Figure 6-21** shows the workpiece under the simulation process. In this process, the semi-cylindrical surface of the workpiece is chosen as the fixed fixturing point because it is located on the V-block. The horizontal clamping force is applied on the right side face of the workpiece, while the vertical clamping forces (two vertical clamping forces) are applied to the rectangular surface (the top face of the workpiece). The process includes the displacement and deformation simulation beside the stress distribution.

After estimating the maximum cutting force F_T, another simulation is conducted on the V-block under the estimated values of the reaction forces $R_1$and R_2 (**Figure 6-22**). This process is important for estimating the standing ability of the V-block under machining operations. In this process, the rectangular surface (the bottom face) of the workpiece is selected as the fixed face as it located on the baseplate. The reaction forces estimated from the 3D body diagram approach (Section 6.4.1) are applied on the V-shaped faces of the V-block. The process also includes the stress distribution, displacement and deformation simulation.

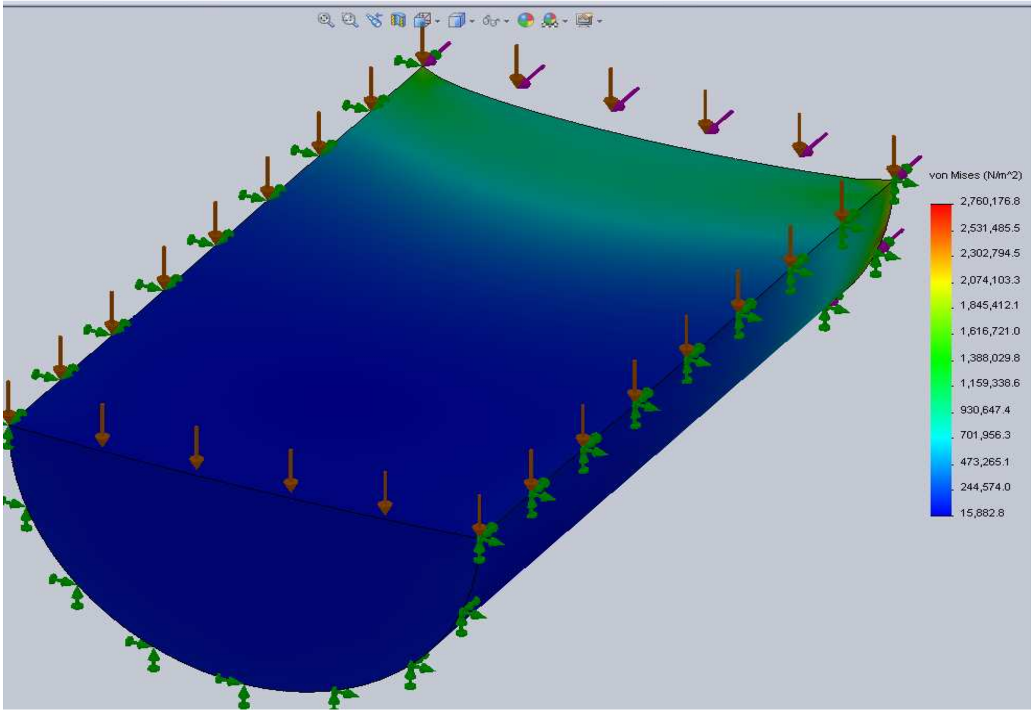


Figure 6-21. The stress distribution of the workpiece under maximum clamping forces.

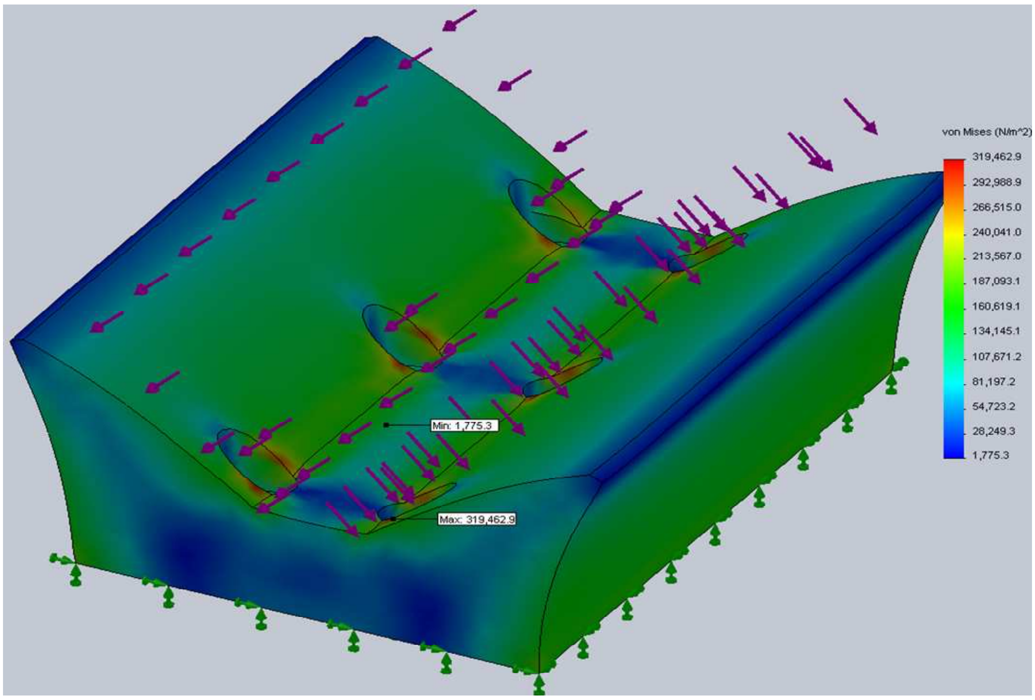


Figure 6-22. The stress simulation of the V-block under reaction forces.

6.5 Results and Discussion

Using the DriveWorksXpress approach results in the generation of new models and configurations for the MF layouts. Although the building of the knowledge base with SolidWorks is a powerful approach for automating the design and assembly process for MF design, this approach experienced minor errors during the knowledge base building process and when performing the generation process. This is related to the limitation of DriveWorksXpress in terms of generating new configurations for the original model. This limitation could be overcome by employing a professional version of SolidWorks with DriveWorksSolo, an improved version of DriveWorksXpress, in order to obtain the maximum benefits from this approach.

Regarding updating the feature library, the user should follow the correct sequence. This sequence begins by creating new folders for the new elements in the Modular Fixtures folder, where the names of these folders should be unique. That means no folders should be created with similar names to those already created in the feature library.

From the results of the cutting force F_T for both the X and Y directions, it is clear that the F_T values are changed. In addition, the values of the reaction force R1 also change with respect to the direction of F_T. This proves that the magnitude of the cutting force changes with direction during the machining operations. From the simulation process, the results show that the maximum stress within the workpiece is 2760176.8 N/m^2 under the maximum clamping forces. This value is within the limit of the maximum values of the workpiece's material stresses, which for grey cast iron is151658000 N/m^2 for the tensile strength and 572165000 N/m^2 for the compressive strength. The same results are found for the V-block under the reaction forces with a maximum stress of 31946.9 N/m^2, which is below the maximum limits as the V-block is made from the same material.

6.6 Summary

An approach for automating the design and assembly process in SolidWorks is illustrated. This approach includes applying DriveWorksXpress and is discussed in terms of generating new models for the fixture components and their assembly layouts. The importance of the feature library for gathering the fixture components is highlighted. The feature library is constructed in two ways. The first way is with the DriveWorksXpress tool itself, as the feature library is generated automatically by capturing the fixture elements and their specifications. The second way is by using the library design features in SolidWorks. The feature library contains the standard fixture components that are derived in SolidWorks file format, as well as components that are newly designed for the purposes of this research. The estimation process of the cutting forces is developed and finite element simulation is applied in order to ensure that the fixturing layout is under the safe considerations for machining operations.

7. Conclusions and Future Work

Developing an integrated modular fixture design and assembly system is the major objective of this thesis. The integration process includes the implementation of four main components; an expert system, SolidWorks, Visual Basic 6 and a feature library of modular fixture elements. SolidWorks is considered as the core of the developed system, and the aim is to integrate the other components with this software. The use of SolidWorks as a powerful CAD software, is due to its excellent 3D modelling capabilities for design and assembly processes. Similarly, in this study Visual Basic 6 is attractive as a programming language because of its specific capabilities to integrate with other software. The expert system has been employed to develop the selection approach for the appropriate fixture elements and fixturing methods (locating and clamping methods). The VisiRule expert system toolkit is implemented for this purpose by creating the knowledge base in an IF-THEN rules format. VisiRule is selected due to its capability to create flowcharts to complete the decision process involved in selecting fixture elements and related methods. Establishing the selection decision process allows the completion of the assembly simulation; for this purpose SolidWorks API is employed with VB6 to create menus in SolidWorks in order to automate the assembly process.

7.1 Research Outcomes

In order to highlight the outcomes of this thesis it is important to address the issues related to the research topic and explain how these issues are addressed in the developed system. These issues are:

- The importance of SolidWorks and how its role can reflect the design and assembly processes of the workpiece and the fixture elements.

CAD software was used in the design process for the MF systems. AutoCAD is the most common CAD software used in previous studies, as explained in Chapter 2. However, most of these studies used AutoCAD with a 2D design approach and for prismatic parts. Moreover, for modelling purposes they applied separate tools for user graphic interface, such as UG-II. Therefore, SolidWorks is employed in this study because of its excellent 3D design environment that can meet the requirements for MF design and assembly. This makes SolidWorks a powerful CAD software that can enhance the design and assembly processes of MF for the semi-circular workpieces that are considered in this research. The implementation of this software significantly simplified the verification of the geometric and machining features of the workpiece.

- Implement the expert system in order to automate the selection process for the locating and clamping methods with MFs to achieve feasible layouts.

Many approaches have been used for selecting the locating and clamping methods and the fixture elements (Chapter 2). These approaches include the use of AI technology such as GA, fuzzy logic and expert systems. However each of these tools can be used so that specific requirements or needs can be integrated with other tools to solve complex cases. Therefore, in order to simplify the selection process, a knowledge base for the selection process has been created and the VisiRule expert system toolkit is used for automating this process. VisiRule has been successfully implemented due to its capabilities for automating the decision making process based on building flowcharts with different features and structures. The knowledge base is built to meet the requirements for the workholding system for a semi-circular workpiece, including side clamping and V-block locating methods.

Based on this knowledge base, flowcharts are developed in VisiRule to generate the programming code for the selection process. This code can be converted to other programming languages such as XML, and can be used with several applications such as Visual Basic. Accordingly, the knowledge base developed can be applied to other kinds of design software. Therefore, VisiRule is considered a comprehensive tool for obtaining the appropriate decision for selecting the locating and clamping methods and their components in order to achieve feasible fixturing layouts for MF.

- The implementation of Visual Basic in terms of automating the assembly process of MF and how integration with SolidWorks could improve this process.

The integration of a programming language with CAD software was used for automating the design and assembly processes of MF. Many methods have been applied for this purpose (Chapter 2). The integration was completed by specific languages and with CAD software such as C++ and AutoCAD. Some studies applied other CAD software such as SolidWorks with C++. Since SolidWorks is employed as a powerful CAD environment with supporting API functions and methods, Visual Basic 6 is applied for programming purposes in this study. This is because VB has the capabilities to build macros in any Microsoft software and can access many functions of API in terms of developing graphical user interfaces. Moreover, VB is considered a flexible programming language that can be integrated with different applications, and it is not expensive compared to other languages. Therefore, the integration of SolidWorks and VB enhances and simplifies the automation process for determining MF layouts.

For assembly automation purposes, a knowledge base is built for inserting the fixture elements and assembling them in the SolidWorks environment. The knowledge base contains the necessary commands and functions in terms of automating the assembly process. Macro functions are also employed for simplifying this process. The results of this integration can be listed as:

1- building new menus in the SolidWorks environment for modular fixture systems;

2- creation of user interfaces for each fixture element to improve the selection and insertion of these elements in SolidWorks, and

3- automation of the assembly of these elements for achieving a feasible fixturing layout.

- The importance of the feature library for completing the integrated system.

For any design and assembly system and process the feature library is a necessary component and a crucial factor for completing these processes. The feature library can be constructed in different ways. In this study, SolidWorks library features are employed for building the feature library for MF elements. The feature library is classified in a way that makes the selection of the fixture components uncomplicated, rapid and efficient, and thus significantly enhances the flexibility and efficiency of the integrated system. The feature library is divided into several classifications, such as the type of clamps, locators, baseplates, supporters and other accessories.

Another approach is investigated in SolidWorks for automating purposes involving the use of the DriveWorksXpress tool. This is used to automate the design and assembly process for determining MF layouts in SolidWorks. Moreover, other SolidWorks tools, such as interference detection and finite elements simulation, are applied in the estimation of the cutting forces, and thus complete the requirement of an integrated system.

7.2 Future Work

The work presented here contributes towards developing an automat CAFD system. Yet, the approach presented in this thesis could be further enhanced and extended by considering the following goals:

- Undertaking a direct and full integration between SolidWorks and VisiRule.

One of the limitations of the integrated approach is the direct integration of VisiRule with SolidWorks. This could be overcome by employing VisiRule functions and features in the SolidWorks API. Then, by including these functions in the knowledge base, the expert system could run automatically in SolidWorks. User interfaces can be also created for this purpose.

- Extending the knowledge base for the selection process to include more MF configurations and elements, other kinds of workpieces, several setups for the selected workpieces and different MF layouts.

The selection approach in this study is completed for semi-circular workpieces to build a workholding system for glass container moulds. However the knowledge base for the selection approach is built in a flexible way and can be extended to other types of workpieces. In this case, other fixture elements should be considered with the appropriate locating and clamping methods. Extra IF-THEN rules could be constructed for this purpose, and more flowcharts could be completed in the VisiRule toolkit to generate the programming code and automate the selection process.

- Creating extended Visual Basic codes in order to make the assembly process more comprehensive and more automated.

More programming codes could be generated to make the assembly process fully automated. The codes can include determining the positions of the fixture elements, the position of the workpiece, and modifying the existing positions of the elements. Moreover, extra codes can be developed for controlling the elements during rotation and movement to meet the requirements of the fixturing layout.

The approach presented in this thesis helps engineers in design and manufacturing fields to simplify the design task of MFs and achieve appropriate layouts in a way that saves production time and costs for the machining operations of variety of workpieces.

8. References

[1] L. Kailing, L. Ran, B. Guiheng, and Z. Peng, "Development of an intelligent jig and fixture design system," *7th International Conference on Computer-Aided Industrial Design and Conceptual Design*, 2006, pp. 1-5.

[2] I. M. Boyle, R. Kevin, and D. C. Brown, "CAFixD: A Case-Based Reasoning Fixture Design Method. Framework and Indexing Mechanisms," *DETC ‘04 ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake,Utah USA, 2004, pp. 1-9.

[3] Y. Rong and X. Li, "Locating method analysis based rapid fxiture configuration design," *IEEE*, pp. 27-32, 1997.

[4] J. Cecil, "A clamping design approach for automated fixture design," *Advanced manufacturing technology*, vol. 18, pp. 784 - 789, 2001.

[5] J. Kopac, J. Balic, and F. Cus, "Sustainable 3D mould design and manufacturing," *PICMET 2008 Proceedings*, Cape Town, South Africa, 2008, pp. 204-209.

[6] F. Mervyn, A. S. Kumar, and A. Y. C. Nee, "Fixture design information support for integrated design and manufacturing," *International journal of production research*, vol. 44, pp. 2205-2219, 2006.

[7] G. Peng, H. Xu, Y. Haiquan, H. Xin, and K. Alipour, "Precise manipulation approach to facilitate interactive modular fixtue assembly design in a virtual environment," *Assembly automation*, vol. 28, pp. 216 - 224, 2008.

[8] H. Tim, "COMPUTERIZING CREATIVITY: THE EVOLUTION OF COMPUTER-AIDED FIXTURING," *Fabricating & Metalworking*, vol. 5, pp. 54-57, 2006.

[9] Kulankara Krishnakumar and S. N. Melkote, "Machining fixture layout optimization using the genetic algorithm," *International journal of machine tools & manufacture*, vol. 40, pp. 579-598, 2000.

[10] W. Chen, L. Ni, and J. Xue, "Deformation control through fixture layout design and clamping force optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 38, pp. 860-867, 2008.

[11] M. Hamedi, "Intelligent Fixture Design through a Hybrid System of Artificial Neural Network and Genetic Algorithm," *Artificial Intelligence Review*, vol. 23, pp. 295-311, 2005.

[12] N. Kaya, "Machining fixture locating and clamping position optimization using genetic algorithms," *Computers in industry*, vol. 57, pp. 112-120, 2006.

[13] K. Kulankara, S. Satyanarayana, and S. N. Melkote, "Iterative Fixture Layout and Clamping Force Optimization Using the Genetic Algorithm," *Journal of manufacturing science and engineering*, vol. 124, pp. 119-125, 2002.

[14] B. N. Saeed, "*An introduction to robotics analysis, systems, applications*": Prentice Hall, 2001.

[15] P. Martin and M. Lombard, "Modelling knowledge related to the allocation of modular jigs for part fixturing using fuzzy reasoning," *The International Journal of Advanced Manufacturing Technology,* vol. 28, pp. 527-531, 2006.

[16] Y. Zhang and G. Peng, "Development of an integrated system for setup planning and fixture design in CAPP," *International conference on advanced intelligent machatronics,* Monterey, California, USA, 2005, pp. 1401-1406.

[17] C. Guangfeng and S. Yize, "Algorithm for workpiece locating design," *1st International Symposium on Systems and Control in Aerospace and Astronautics*, 2006, pp. 1462-1467.

[18] Y. G. Kang, Z. Wang, R. Li, and C. Jiang, "A fixture design system for networked manufacturing," *International journal of computer integrated manufacturing,* vol. 20, pp. 143-159, 2007.

[19] C. S. Krishanmoorthy and S. Rajeev, "*Artificial Intelligence and Expert Systems for Engineers*". New York: CRC Press, Inc, 1996.

[20] A. S. Kumar, A. Y. C. Nee, and S. Prombanpong, "Expert fixture-design system for an automated manufacturing environment," *Computer-Aided Design,* vol. 24, pp. 316-326, 1992.

[21] A. Y. C. Nee and A. S. kumar, "A Framework for an Object/Rule-Based Automated Fixture Design System," *CIRP Annals - Manufacturing Technology,* vol. 40, pp. 147-151, 1991.

[22] B. O. Nnaji and S. Alladin, "E-CAFFS: An expert computer-aided flexible fixturing system," *Computers & Industrial Engineering,* vol. 18, pp. 297-311, 1990.

[23] B. S. Babu, P. M. Valli, A. V. V. Kumar, and D. Rao, "Automatic modular fixture generation in computer aided process planning systems," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* pp. 1147-1152, 2005.

[24] T. S. Kow, A. S. Kumar, and J. Y. H. Fuh, "An Integrated Approach to Collision-Free Computer-Aided Modular Fixture Design," *International Journal of Advanced Manufacturing Technology,* vol. 16, pp. 233-242, 2000.

[25] W. Ma, Z. Lei, and Y. Rong, "FIX-DES:A computer - aided modular fixture configuration design system," *Advanced manufacturing technology,* vol. 14, pp. 21-32, 1998.

[26] X. Kong, Yangyi, J. Zhou, C. Gou, H. Zhang, and W. Zhao, "Reaserch and development of the software on computer aided fixtures designing," *IEEE,* pp. 1233-1236, 2009.

[27] M. Shokri and B. Arezoo, "Computer-aided CMM modular fixture configuration design," *International Journal of Manufacturing Technology and Managment,* vol. 14, pp. 174-187, 2008.

[28] J. L. Hou and A. J. C. Trappey, "Computer- aided fixture design system for comprehensive modular fixtures," *International journal of production research,* vol. 39, pp. 3703-3725, 2001.

[29] J. R. Dai, A. Y. C. Nee, J. Y. H. Fuh, and A. S. Kumar, "An approach to automating modular fixture design and assembly," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 211, pp. 509-521, 1997.

[30] L. B. Siong, T. Imao, H. Yoshida, K. Goto, S. L. Koh, D. Lim, L. Chin, and S. C. Gan, "Integrated modular fixture design, pricing and inventory control expert system," *international journal of production research,* vol. 30, pp. 2019-2044, 1992.

[31] A. Joneja and T.-C. Chang, "Setup and fixture planning in automated process planning systems," *IIE Transactions,* vol. 31, pp. 653-665, 1999.

[32] K. Castelino, V. Sundararajan, R. D'Souza, B. Kannan, and P. K. Wright, "AMPS-An Automated Modular Process Planning System," *Journal of Computing and Information Science in Engineering,* vol. 4, pp. 235-241, 2004.

[33] T. C. Chang, R. A. Wysk, and H. P. Wang, "*Computer aided manufacturing*", 3 ed. New Jersey: Pearson Education Inc, 2006.

[34] E. G. Hoffman, "*Jig and fixture design*", 5 ed. New York: Delmar Learning Drafting Series, 2004.

[35] E. K. Henriksen, "*Jig and fixture design manual*". New York: Industrial Press Inc, 1973.

[36] J. T. Black and R. A. Kohser, "*Materials and processes in manufacturing*", 10 ed. New Jersey: Wiley and Sons, Inc., 2008.

[37] M. Krsulia, B. Barisic, and J. Kudlacek, "Assembly setup for modular fixture machining process," *Advanced Engineering,* vol. 3, pp. 39-51, 2009.

[38] *Locating and Clamping Principles*. Available: http://www.carrlane.com/Catalog/index.cfm/29625071F0B221118070C1C513906103E0B05543B0B012009083C3B285351405948201318 0B041D1E173C3B2853524B5A59

[39] *KIPP WORKHOLDING SYSTEMS*. Available: http://www.kipp.com/App/WebObjects/XSeMIPSKIPP.woa/cms/page/locale.enGB/pid.1097.1105.1227/Workholding-systems.html

[40] G. Peng, G. Jun, and H. Xu, "Towards the development of a desktop virtual reality- based system for modular fixture congiguration design," *Assembly automation,* vol. 29, pp. 19-31, 2009.

[41] *Computer-aided Modular Fixture Configuration Design Principle*. Available: http://www.me.wpi.edu/research/CAMLab/research%20areas/html/principl.htm

[42] Y. Bai and K. Rong, "Modular fixture element modeling and assembly relationship analysis for automated fixture configuration design," *Engineering design and automation,* vol. 4, pp. 147-162, 1998.

[43] G. Peng, G. Wang, and Y. Chen, "A pragmatic system to support interactive modular fixture configuration design in desktop virtual environment," *IEEE*, vol. 2, pp. 19-24, 2008.

[44] U. H. Farhan and M. T. Rad, "Design of modular fixtures using a 3D-modelling approach," *19th International Congress on Modelling and Simulation*, Perth, Australia, 2011, pp. 405-411.

[45] *Development of Automated Dedicated Fixture Design Systems*. Available: http://www.me.wpi.edu/research/CAMLab/research%20areas/new/ded_principle.htm

[46] G. C. Burdea and P. Coiffet, "*Virtual reality technology*". New Jersey: Wiley, 2003.

[47] M. C. Kerman and R. L. Brown, "*Computer programming fundamentals with applications in visual basic 6.0*". Reading, Mass: Addison-Wesley, 2000.

[48] H. M. Deitel, P. J. Deitel, and T. R. Nieto, "*Visual Basic 6 how to program*". Upper Saddle River, N.J: Prentice Hall, 1999.

[49] S. P. Prince, R. G. Ryan, and T. Mincer, "Common API : Using Visual Basic to Communicate between Engineering Design and Analytical Software Tools," *ASEE Annual Conferenc*, 2005.

[50] M. C. Doo, J. K. Hyung, C. L. Jae, and H. A. Sung, "Web-Based Material Database for Material Selection and its Application Programming Interface (API) for CAD," *Key Engineering Materials*, vol. 345-346, pp. 1593-1596, 2007.

[51] S. Bo, Q. Guangtai, and F. Yadong, "Research of standard parts library construction for SolidWorks by Visual Basic," *Electronic and Mechanical Engineering and Information Technology (EMEIT) International Conference*, 2011, pp. 2651-2654.

[52] Y. Peng, J. Xie, and X. Wang, "Research and Realization on Architectural 3D Model of Architectural Process Simulation System," *Second International Workshop on computer science and engineering*, 2009, pp. 543-547.

[53] X. Ning and Q. Jiang, "A digital design method of geometric model for centrifugal fan impeller based on SolidWorks and VB," *International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT)*, 2011, pp. 4023-4026.

[54] S. Danjou, N. Lupa, and P. Koehler, "Approach for Automated Product Modeling Using Knowledge-Based Design Features," *Computer-Aided Design and Applications*, vol. 5, pp. 622-629, 2008.

[55] J. Tian, S. Liu, and H. Fu, "CAD System Design on Standard Part Based on Software Reuse," *Fourth International Symposium on Knowledge Acquisition and Modeling (KAM)*, 2011, pp. 229-232.

[56] M. Zhen and J. Yingyi, "Automatic assembly for combined mold components based on SolidWorks," *International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 166-169.

[57] Y. Yang, "The parametric design and intelligent assembly system based on the secondary development of solidworks," *2nd International Conference on Computer Engineering and Technology (ICCET)*, 2010, pp. 602-605.

[58] *Logic Programming Associates Ltd / Version 4.920 Documentation Files*. Available: http://www.lpa.co.uk/dow_doc.htm

[59] *CarrLane pdf Catalog*. Available: http://www.carrlane.com/pdfcatalog/

[60] *DriveWorksXpress - Design Automation for SolidWorks - Training*. Available: http://www.driveworksxpress.com/index.php/training

[61] S. R. Chavali, C. Sen, G. M. Mocko, and J. D. Summers, "Using Rule Based Design in Engineer to Order Industry: An SME Case Study," *Computer-Aided Design and Applications*, vol. 5, pp. 178-193, 2008.

[62] I. Edmund, "*Engineering formulas for metalcutting:Presented in customary U.S. and metric units of measure*": Industrial Press Inc, 2004.

[63] M. T. Rad, "Studies on CAD/CAM Integration for Milling Operations using Optimum Machining Parameters," Doctor of Philosophy thesis, Faculty of Engineering and the Environment, University of South Australia, 1997.

[64] H. L. Wen, "Modeling of cutting forces in end milling operations," *Tamkang Journal of Science and Engineering*, vol. 3, pp. 15-22, 2000.

9. APPENDICES

Appendix 1: The Knowledge Base Rules for the Selection Process

Rule 001

If the workpiece is semi-circular, *and* the size of the part is within specified limits, *then* modular fixture design is used for defining the locating, clamping and baseplate elements.

Rule 002

If the workpiece is semi-circular and modular fixture design is used, *then* the active locating surface is defined. This kind of workpiece has four surfaces; rectangular surface, semi-cylindrical surface and two semi-circular surfaces. Defining the active locating surface is important for determining the locating method and completing the rest of the process.

Rule 003

If the active locating surface is the semi-cylindrical surface of the workpiece, *then* the V-Blocks are used for locating. In this case, the rectangular surface will be machined and two clamping methods can be considered, top clamping and side clamping.

Rule 004

If the locating surface is not the semi-circular surface of the workpiece, *then* the other surfaces will be considered as active locating surfaces. This depends on which machining operations will be applied to the workpiece. In this research, the rule 3 is applied for the integrated system. The fixturing system for other surfaces could be considered for the future work.

Rule 005

If the workpiece is semi-circular, and the locating method is V-Blocks, *then* the workpiece's radius is calculated. This is important in order to identify the kind of V-Blocks that needed for locating.

Rule 006

If the workpiece's radius is more than or equal to 9 mm and smaller than 25.4 mm, *then* small mini V-Blocks are used. These V-Blocks are appropriate for this range of diameters, and they have two holes for mounting them on the baseplate.

Rule 007

If the workpiece's radius is more than 14 mm and smaller than 50.8 mm, *then* large mini V-Blocks are used. These V-Blocks are suitable for this range of parts, and they have four holes for mounting on the baseplate.

Rule 008

If the workpiece's radius is more than 14 mm and smaller than 102 mm, *then* standard V-Blocks are used. These V-Blocks are proper for small and large diameters, and they have two holes for mounting on the baseplate.

Rule 009

If the workpiece's radius is more than 102 mm, *then* a V-Cast section is used. In this research, a V-Cast section was designed by SolidWorks to meet the requirements of the specific workpiece and all the specifications were considered according to Carrlane catalogues.

Rule 010

If the workpiece is semi-circular *and* the rectangular surface is machined, *then* the machined area of this surface is calculated. This is important to define the clamping method.

Rule 011

If the whole rectangular surface is machined, *then* the side clamping is used. This clamping method allows holding the workpiece with keeping the surface clear for machining.

Rule 012

If the machined area is equal to 50% or less than the whole area of the rectangular surface, *then* top clamping could be used. This depends on the shape and the location of the pattern which should be machined on the surface.

Rule 013

If the side clamping is used, *then* the kind of clamp is identified. For side clamping, three kinds of clamps can be used based on Carrlane company catalogue, pivoting edge clamps, cam edge clamps, and serrated edge clamps. In this research, pivoting edge clamps were considered.

Rule 014

If the top clamping is used, *then* different kind of clamps is used. Top clamping and its related clamps could be considered for future work.

Rule 015

If the pivoting edge clamps are used, *then* the workpiece height is calculated. For semi-circular workpiece with the machined rectangular surface, the height will be equal to the radius of the side surface of the workpiece.

Rule 016

If the workpiece height is equal to or less than 100 mm, *and* the baseplate is standard, *then* standard pivoting clamp and downthrust are used with no riser. *Else*, heavy pivoting clamp and downthrust are used.

Rule 017

If the workpiece height is between 100 mm and 125 mm, *and* the baseplate is standard, *then* standard pivoting clamp and downthrust are used with 25.4 mm riser. *Else*, heavy pivoting clamp and downthrust are used.

Rule 018

If the workpiece height is between 125 mm and 150 mm, *and* the baseplate is standard, *then* standard pivoting clamp and downthrust are used with 50.8 mm riser. *Else*, heavy pivoting clamp and downthrust are used.

Rule 019

If the height is not equal to the radius of the workpiece, *then* the process is stoped and a different set-up of clamping is used. The specification of pivoting clamps, downthrust, and the risers were considered according to the Carrlane catalogue.

Rule 020

If the modular fixture (MF) design is used, *then* the kind of MF system is defined. There are two MFs systems, Hole-dowel and T-slot. In this research, Hole-dowel was considered.

Rule 021

If the Hole- dowel system is used, *and* the side clamping is applied with using pivoting edge clamps, *then* the area of the rectangular surface which should be machined of the workpiece is calculated.

Rule 022

If the machining area is less than or equal to 0.094 m^2, *and* the mini V-blocks are used, *then* 400 mm mini baseplate is used.

Rule 023

If the machining area is less than or equal to 0.094 m², *and* the standard V-blocks are used, *then* 400 mm standard baseplate is used.

Rule 024

If the machining area is between 0.094 m² and 0.0146 m², *and* the standard V-blocks are used, *then* 500 mm standard baseplate is used.

Rule 025

If the machining area is between 0.094 m² and 0.0146 m², *and* the heavy V-blocks are used, *then* 500 mm heavy baseplate is used.

Rule 026

If the machining area is between 0.0146 m² and 0.0232 m², *and* the standard V-blocks are used, *then* 630 mm standard baseplate is used.

Rule 027

If the machining area is between 0.0146 m² and 0.0232 m², *and* the heavy V-blocks are used, *then* 630 mm heavy baseplate is used.

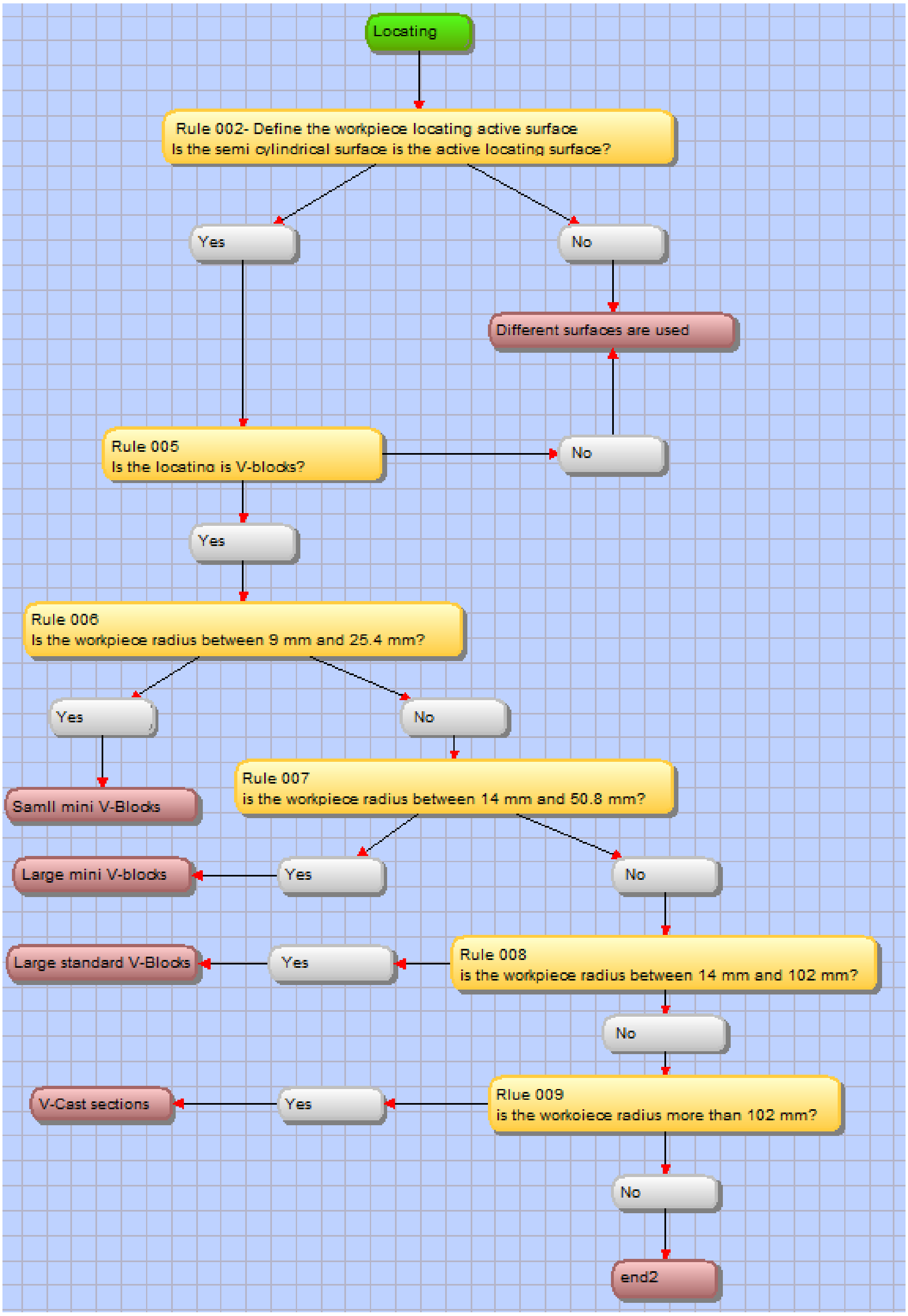
Rule 028

If the machining area is between 0.0232 m² and 0.0375 m², *and* the standard V-blocks are used, *then* 800 mm standard baseplate is used.

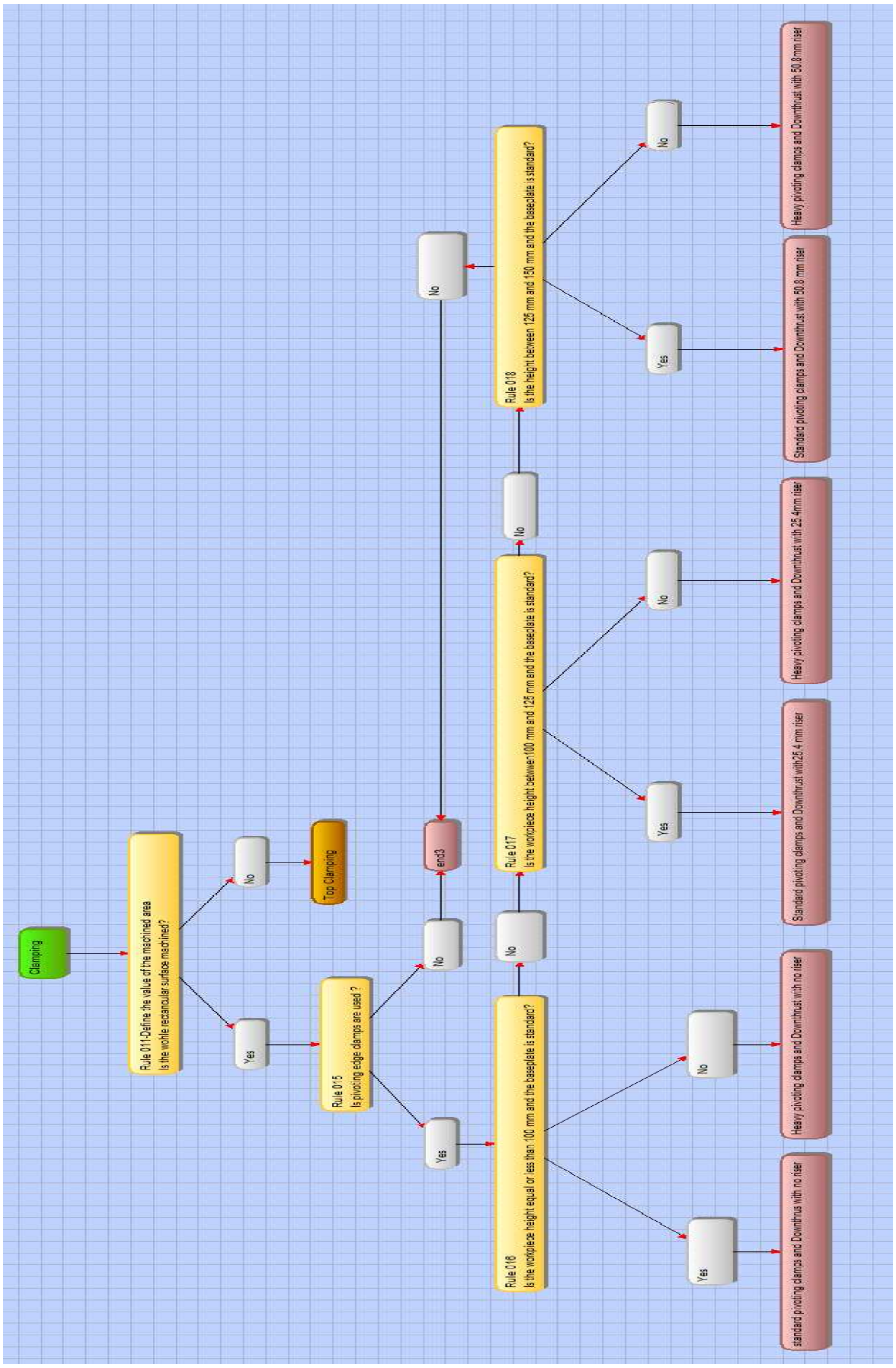
Rule 029

If the machining area is between 0.0232 m² and 0.0375 m², *and* the heavy V-blocks are used, *then* 800 mm heavy baseplate is used.

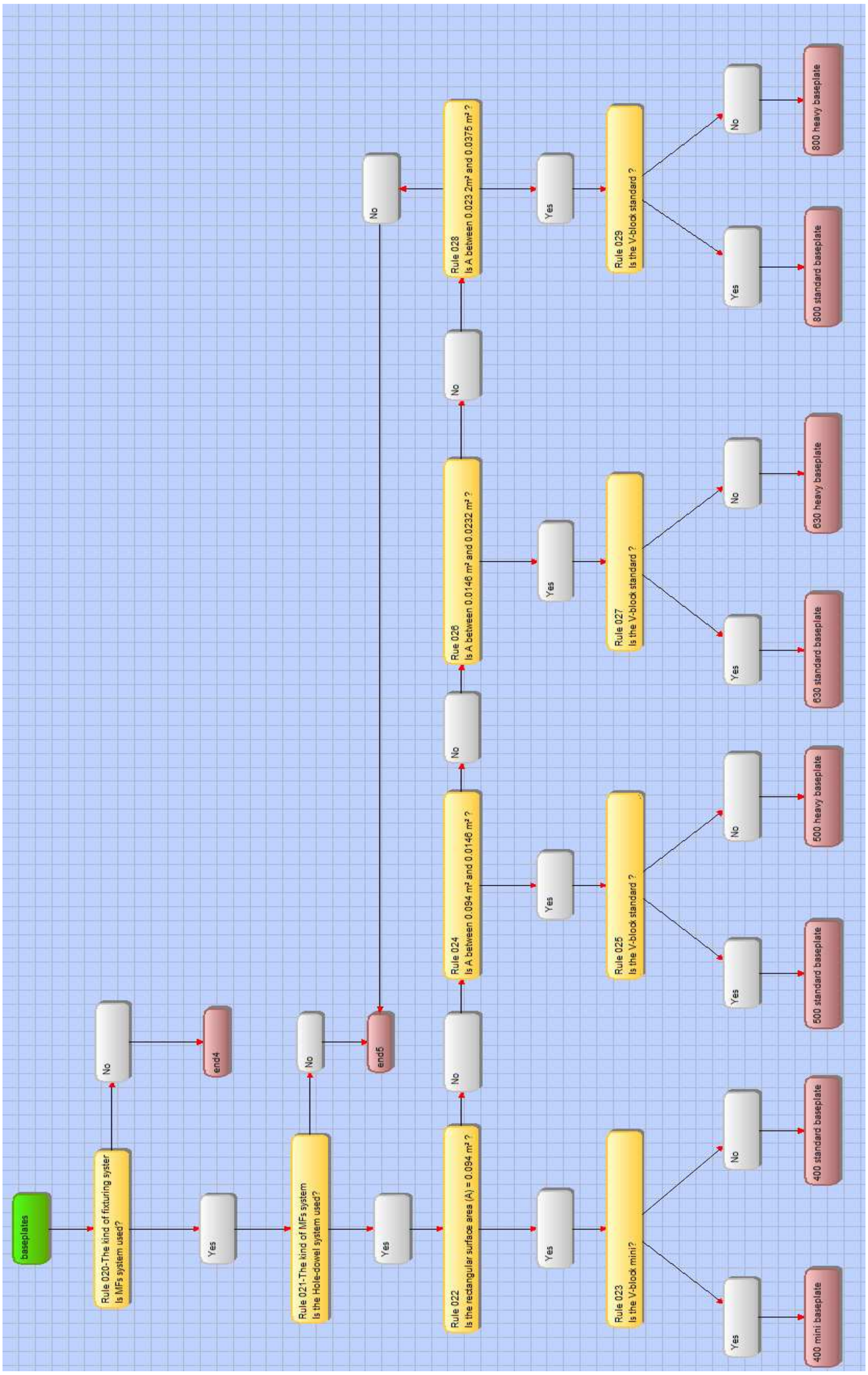
Appendix 2: VisiRule Chart for Locating Method



Appendix 3: VisiRule Chart for Clamping Method



Appendix 4: VisiRule Chart for Baseplate



Appendix 5: The Developed Flex Code for the Selection Process

```
do ensure_loaded( system(vrlib) ) .

relation 'Locating'( Conclusion ) if
   'q_Define the workpiece locating active surface'( Conclusion ) .

relation 'q_Define the workpiece locating active surface'(
Conclusion ) if
   the answer to 'Define the workpiece locating active surface' is _
and
   check( 'Define the workpiece locating active surface', =, 'Yes' )
and
   'q_Rule1 '( Conclusion ) .

relation 'q_Define the workpiece locating active surface'(
Conclusion ) if
   the answer to 'Define the workpiece locating active surface' is _
and
   check( 'Define the workpiece locating active surface', =, 'No' )
and
   Conclusion = 'Different surfaces are used'  .

relation 'q_Rule1 '( Conclusion ) if
   the answer to 'Rule1 ' is _ and
   check( 'Rule1 ', =, 'Yes' ) and
   Conclusion = 'Samll mini V-Blocks '  .

relation 'q_Rule1 '( Conclusion ) if
   the answer to 'Rule1 ' is _ and
   check( 'Rule1 ', =, 'No' ) and
   q_Rule2( Conclusion ) .

relation q_Rule2( Conclusion ) if
   the answer to 'Rule2' is _ and
   check( 'Rule2', =, 'Yes' ) and
   Conclusion = 'Large mini V-blocks'  .

relation q_Rule2( Conclusion ) if
   the answer to 'Rule2' is _ and
   check( 'Rule2', =, 'No' ) and
   q_Rule3( Conclusion ) .

relation q_Rule3( Conclusion ) if
   the answer to 'Rule3' is _ and
   check( 'Rule3', =, 'Yes' ) and
   Conclusion = 'Large standard V-Blocks'  .

relation q_Rule3( Conclusion ) if
   the answer to 'Rule3' is _ and
   check( 'Rule3', =, 'No' ) and
   q_Rlue4( Conclusion ) .

relation q_Rlue4( Conclusion ) if
```

```
   the answer to 'Rlue4' is _ and
   check( 'Rlue4', =, 'Yes' ) and
   Conclusion = 'V-Cast sections'  .

relation q_Rlue4( Conclusion ) if
   the answer to 'Rlue4' is _ and
   check( 'Rlue4', =, 'No' ) and
   Conclusion = end2  .

relation start( Conclusion ) if
   'q_The kind of the workpiece'( Conclusion ) .

relation 'q_The kind of the workpiece'( Conclusion ) if
   the answer to 'The kind of the workpiece' is _ and
   check( 'The kind of the workpiece', =, 'Yes' ) and
   'q_The kind of fixtur design'( Conclusion ) .

relation 'q_The kind of the workpiece'( Conclusion ) if
   the answer to 'The kind of the workpiece' is _ and
   check( 'The kind of the workpiece', =, 'No' ) and
   Conclusion = 'Use different set-up '  .

relation 'q_The kind of fixtur design'( Conclusion ) if
   the answer to 'The kind of fixtur design' is _ and
   check( 'The kind of fixtur design', =, 'Yes' ) and
   'Locating'( Conclusion ) .

relation 'q_The kind of fixtur design'( Conclusion ) if
   the answer to 'The kind of fixtur design' is _ and
   check( 'The kind of fixtur design', =, 'Yes' ) and
   'Clamping'( Conclusion ) .

relation 'q_The kind of fixtur design'( Conclusion ) if
   the answer to 'The kind of fixtur design' is _ and
   check( 'The kind of fixtur design', =, 'Yes' ) and
   'Baseplates'( Conclusion ) .

relation 'q_The kind of fixtur design'( Conclusion ) if
   the answer to 'The kind of fixtur design' is _ and
   check( 'The kind of fixtur design', =, 'No' ) and
   Conclusion = end1  .

relation 'Clamping'( Conclusion ) if
   'q_Define the value of the machined area'( Conclusion ) .

relation 'q_Define the value of the machined area'( Conclusion ) if
   the answer to 'Define the value of the machined area' is _ and
   check( 'Define the value of the machined area', =, 'Yes' ) and
   q_Rule5( Conclusion ) .

relation 'q_Define the value of the machined area'( Conclusion ) if
   the answer to 'Define the value of the machined area' is _ and
   check( 'Define the value of the machined area', =, 'No' ) and
   'Top Clamping '( Conclusion ) .

relation q_Rule5( Conclusion ) if
   the answer to 'Rule5' is _ and
```

```
   check( 'Rule5', =, 'Yes' ) and
   'q_Rule6 '( Conclusion ) .

relation q_Rule5( Conclusion ) if
   the answer to 'Rule5' is _ and
   check( 'Rule5', =, 'No' ) and
   Conclusion = end3  .

relation 'q_Rule6 '( Conclusion ) if
   the answer to 'Rule6 ' is _ and
   check( 'Rule6 ', =, 'Yes' ) and
   q_Rule9( Conclusion ) .

relation 'q_Rule6 '( Conclusion ) if
   the answer to 'Rule6 ' is _ and
   check( 'Rule6 ', =, 'No' ) and
   q_Rule7( Conclusion ) .

relation q_Rule9( Conclusion ) if
   the answer to 'Rule9' is _ and
   check( 'Rule9', =, 'Yes' ) and
   Conclusion = 'standard pivoting clamps and Downthrus with no
riser'  .

relation q_Rule9( Conclusion ) if
   the answer to 'Rule9' is _ and
   check( 'Rule9', =, 'No' ) and
   Conclusion = 'Heavy pivoting clamps and Downthrust with no riser'
.

relation q_Rule7( Conclusion ) if
   the answer to 'Rule7' is _ and
   check( 'Rule7', =, 'Yes' ) and
   q_Rule10( Conclusion ) .

relation q_Rule7( Conclusion ) if
   the answer to 'Rule7' is _ and
   check( 'Rule7', =, 'No' ) and
   q_Rule8( Conclusion ) .

relation q_Rule10( Conclusion ) if
   the answer to 'Rule10' is _ and
   check( 'Rule10', =, 'Yes' ) and
   Conclusion = 'Standard pivoting clamps and Downthrust with25.4 mm
riser'  .

relation q_Rule10( Conclusion ) if
   the answer to 'Rule10' is _ and
   check( 'Rule10', =, 'No' ) and
   Conclusion = 'Heavy pivoting clamps and Downthrust with 25.4mm
riser'  .

relation q_Rule8( Conclusion ) if
   the answer to 'Rule8' is _ and
   check( 'Rule8', =, 'Yes' ) and
   q_Rule11( Conclusion ) .
```

```
relation q_Rule8( Conclusion ) if
   the answer to 'Rule8' is _ and
   check( 'Rule8', =, 'No' ) and
   Conclusion = end3  .

relation q_Rule11( Conclusion ) if
   the answer to 'Rule11' is _ and
   check( 'Rule11', =, 'Yes' ) and
   Conclusion = 'Standard pivoting clamps and Downthrust with 50.8
mm riser'  .

relation q_Rule11( Conclusion ) if
   the answer to 'Rule11' is _ and
   check( 'Rule11', =, 'No' ) and
   Conclusion = 'Heavy pivoting clamps and Downthrust with 50.8mm
riser'  .

relation baseplates( Conclusion ) if
   'q_The kind of fixturing system'( Conclusion ) .

relation 'q_The kind of fixturing system'( Conclusion ) if
   the answer to 'The kind of fixturing system' is _ and
   check( 'The kind of fixturing system', =, 'Yes' ) and
   'q_The kind of MFs system'( Conclusion ) .

relation 'q_The kind of fixturing system'( Conclusion ) if
   the answer to 'The kind of fixturing system' is _ and
   check( 'The kind of fixturing system', =, 'No' ) and
   Conclusion = end4  .

relation 'q_The kind of MFs system'( Conclusion ) if
   the answer to 'The kind of MFs system' is _ and
   check( 'The kind of MFs system', =, 'Yes' ) and
   'q_Rule 12'( Conclusion ) .

relation 'q_The kind of MFs system'( Conclusion ) if
   the answer to 'The kind of MFs system' is _ and
   check( 'The kind of MFs system', =, 'No' ) and
   Conclusion = end5  .

relation 'q_Rule 12'( Conclusion ) if
   the answer to 'Rule 12' is _ and
   check( 'Rule 12', =, 'Yes' ) and
   'q_Rule 13'( Conclusion ) .

relation 'q_Rule 12'( Conclusion ) if
   the answer to 'Rule 12' is _ and
   check( 'Rule 12', =, 'No' ) and
   'q_Rule 14'( Conclusion ) .

relation 'q_Rule 13'( Conclusion ) if
   the answer to 'Rule 13' is _ and
   check( 'Rule 13', =, 'Yes' ) and
   Conclusion = '400 mini baseplate'  .

relation 'q_Rule 13'( Conclusion ) if
```

```
   the answer to 'Rule 13' is _ and
   check( 'Rule 13', =, 'No' ) and
   Conclusion = '400 standard baseplate'  .

relation 'q_Rule 14'( Conclusion ) if
   the answer to 'Rule 14' is _ and
   check( 'Rule 14', =, 'Yes' ) and
   'q_Rule 15'( Conclusion ) .

relation 'q_Rule 14'( Conclusion ) if
   the answer to 'Rule 14' is _ and
   check( 'Rule 14', =, 'No' ) and
   'q_Rule 16'( Conclusion ) .

relation 'q_Rule 15'( Conclusion ) if
   the answer to 'Rule 15' is _ and
   check( 'Rule 15', =, 'Yes' ) and
   Conclusion = '500 standard baseplate'  .

relation 'q_Rule 15'( Conclusion ) if
   the answer to 'Rule 15' is _ and
   check( 'Rule 15', =, 'No' ) and
   Conclusion = '500 heavy baseplate'  .

relation 'q_Rule 16'( Conclusion ) if
   the answer to 'Rule 16' is _ and
   check( 'Rule 16', =, 'Yes' ) and
   'q_Rule 17'( Conclusion ) .

relation 'q_Rule 16'( Conclusion ) if
   the answer to 'Rule 16' is _ and
   check( 'Rule 16', =, 'No' ) and
   'q_Rule 18'( Conclusion ) .

relation 'q_Rule 17'( Conclusion ) if
   the answer to 'Rule 17' is _ and
   check( 'Rule 17', =, 'Yes' ) and
   Conclusion = '630 standard baseplate'  .

relation 'q_Rule 17'( Conclusion ) if
   the answer to 'Rule 17' is _ and
   check( 'Rule 17', =, 'No' ) and
   Conclusion = '630 heavy baseplate'  .

relation 'q_Rule 18'( Conclusion ) if
   the answer to 'Rule 18' is _ and
   check( 'Rule 18', =, 'No' ) and
   Conclusion = end5  .

relation 'q_Rule 18'( Conclusion ) if
   the answer to 'Rule 18' is _ and
   check( 'Rule 18', =, 'Yes' ) and
   'q_Rule 19'( Conclusion ) .

relation 'q_Rule 19'( Conclusion ) if
   the answer to 'Rule 19' is _ and
   check( 'Rule 19', =, 'Yes' ) and
```

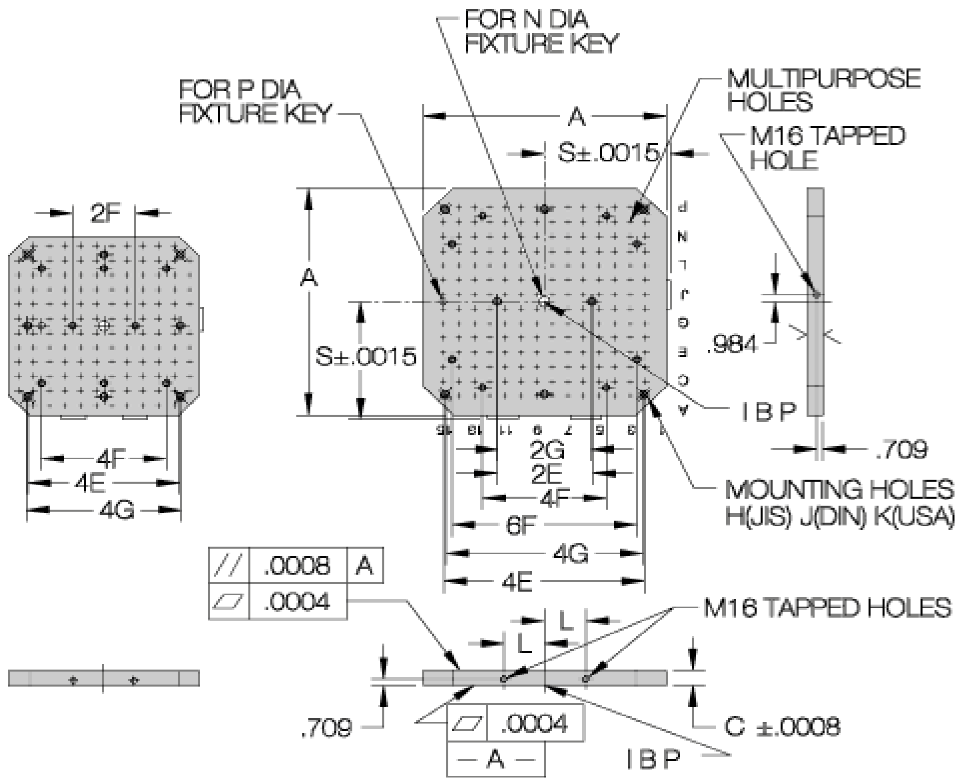
```
   Conclusion = '800 standard baseplate'  .

relation 'q_Rule 19'( Conclusion ) if
   the answer to 'Rule 19' is _ and
   check( 'Rule 19', =, 'No' ) and
   Conclusion = '800 heavy baseplate'  .

group group1
   'Yes', 'No' .

question 'Rlue4'
   'is the radius more than 102mm?' ;
   choose one of group1
   because '' .

question 'Rule3'
   'is the radius between 14 and 102mm?' ;
   choose one of group1
   because '' .

question 'Rule2'
   'is the workpiece radius between 14 and 50.8mm? ' ;
   choose one of group1
   because '' .

question 'Define the workpiece locating active surface'
   'Is the semi cylindrical surface is the active locating surface?'
;
   choose one of group1
   because '' .

question 'Rule1 '
   'Is the workpiece radius between 9 and 25.4 mm? ' ;
   choose one of group1
   because '' .

question 'The kind of the workpiece'
   'Is the workpiece semi-circular?' ;
   choose one of group1
   because '' .

question 'The kind of fixtur design'
   'Is Modular fixture is used? ' ;
   choose one of group1
   because '' .

question 'Define the value of the machined area'
   'is the all rectancular surface is machined?' ;
   choose one of group1
   because '' .

question 'Rule5'
   'is pivpting edge clamps are used ? ' ;
   choose one of group1
   because '' .

question 'Rule9'
```

```
   'is the baseplae used standard? ' ;
   choose one of group1
   because '' .

question 'Rule6 '
   'is the workpiece height is equal or less than 100mm?' ;
   choose one of group1
   because '' .

question 'Rule11'
   'is the baseplate standarad?' ;
   choose one of group1
   because '' .

question 'Rule8'
   'is the the height between 125mm and 150mm?' ;
   choose one of group1
   because explanation .

question 'Rule10'
   'is the baseplate standard?' ;
   choose one of group1
   because '' .

question 'Rule7'
   'is the workpiece height is betwwen 100mm and 125mm?' ;
   choose one of group1
   because '' .

question 'Rule 19'
   'Is the V-blocks are standard ?' ;
   choose one of group1
   because '' .

question 'Rule 15'
   'Is the V-blocks are standard ?' ;
   choose one of group1
   because '' .

question 'Rule 14'
   'Is A between 0.094m² and 0.0146m² ?' ;
   choose one of group1
   because '' .

group group2
   'No', 'Yes' .

question 'Rule 18'
   'Is A between 0.0232m² and 0.0375m² ?' ;
   choose one of group2
   because '' .

question 'Rule 17'
   'Is the V-blocks are standard ?' ;
   choose one of group1
   because '' .
```

```
question 'Rule 16'
   'Is A between 0.0146m² and 0.0232m² ?' ;
   choose one of group1
   because '' .

question 'The kind of fixturing system'
   'Is MFs system used?' ;
   choose one of group1
   because '' .

question 'The kind of MFs system'
   'Is the Hole-dowel system used?' ;
   choose one of group1
   because '' .

question 'Rule 12'
   'Is the rectangular surface area(A) = 0.094m² ? ' ;
   choose one of group1
   because '' .

question 'Rule 13'
   'Is the V-blocks are mini?' ;
   choose one of group1
   because '' .
```

Appendix 6: The Baseplates Specifications

Gray Cast Iron, ASTM CLASS 40

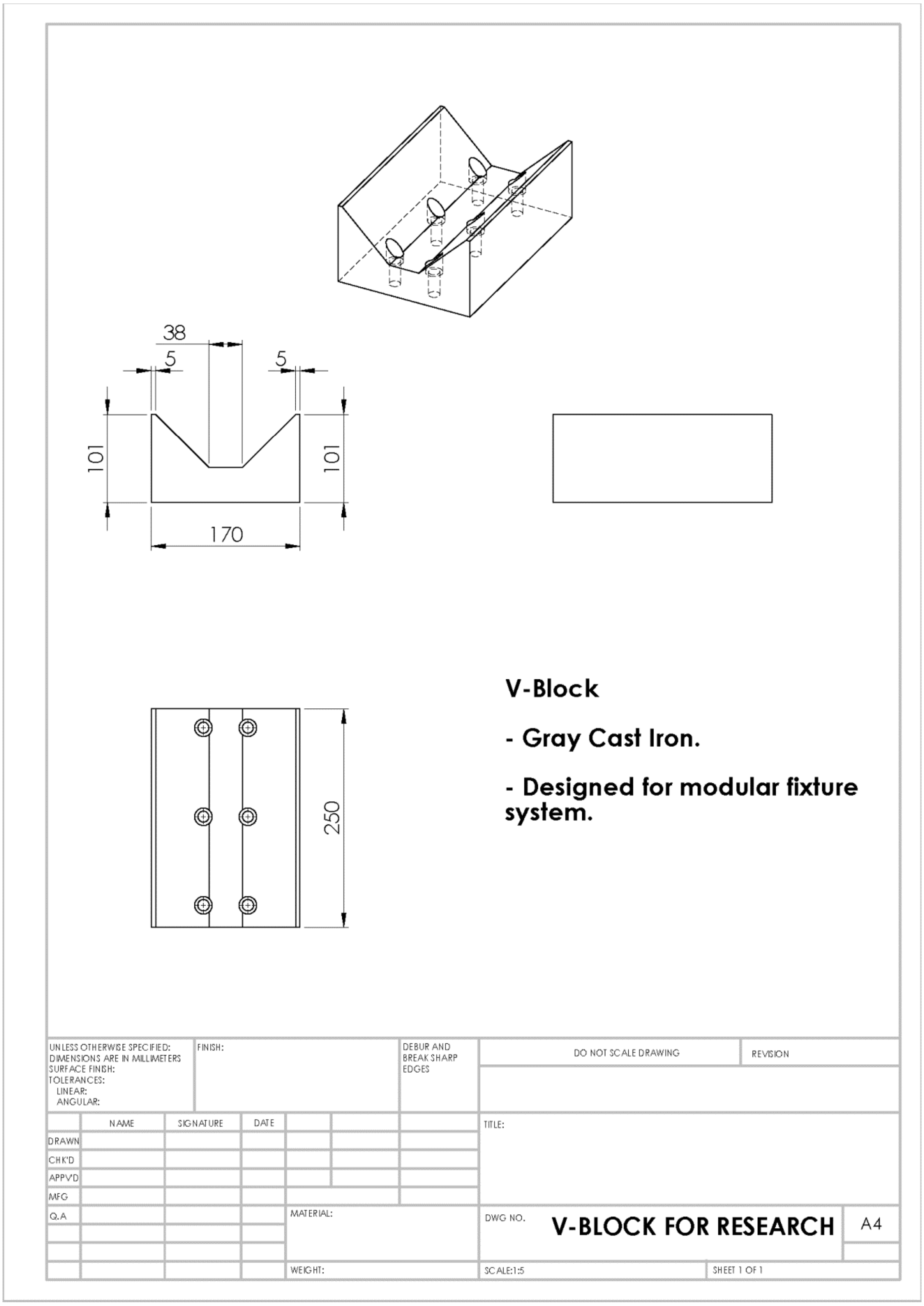


STANDARD

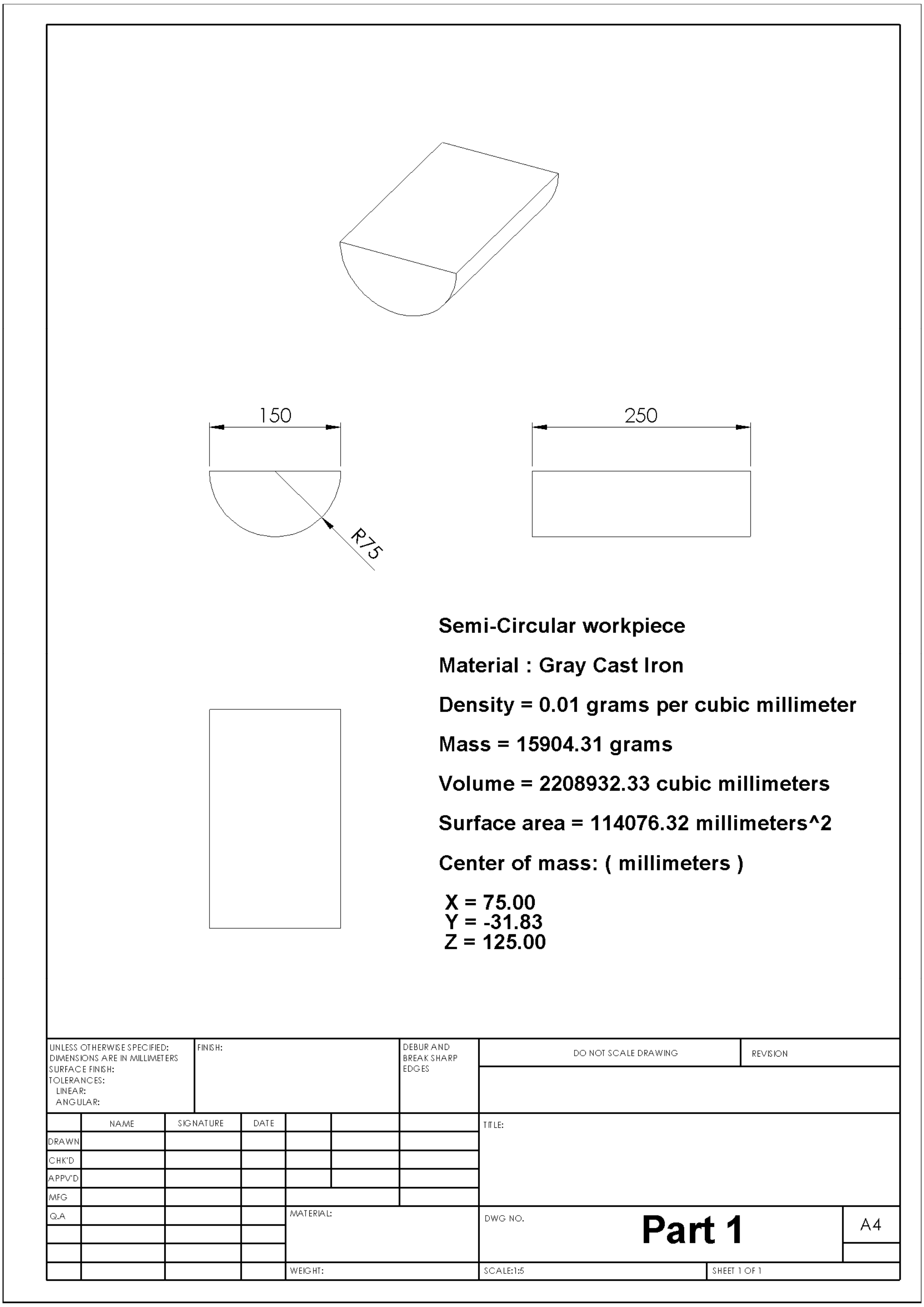
PART NO.		CL-MF40-0801	CL-MF40-0802	CL-MF40-0803	CL-MF40-0804*
PALLET SIZE		400mm	500mm	630mm	800mm
MULTIPURPOSE HOLES	THREAD	1/2"-13			
	ID	.5000"			
	SPACING	2.0000"			
	PATTERN	8x8	10x10	12x12	14x14
A	NOMINAL	400mm	500mm	630mm	800mm
	ACTUAL	386mm	486mm	616mm	786mm
C		2.0000"		2.0000"	2.3750"
E (JIS)		80mm	100mm	125mm	160mm
F (DIN)		50mm	100mm	100mm	
G (USA)		3.000"	4.000"	5.000"	6.000"
H (JIS)		M16			
J (DIN)		M12		M16	
K (USA)		5/8"-11 or M16			
L		55mm	75mm	100mm	135mm
N DIA		30mm			
P DIA		20mm		25mm	
S		200mm	250mm	315mm	400mm
WEIGHT (KG)		53	82	132	255

CarrLane MANUFACTURING CO.

Appendix 7: The Designed V-Block Specifications

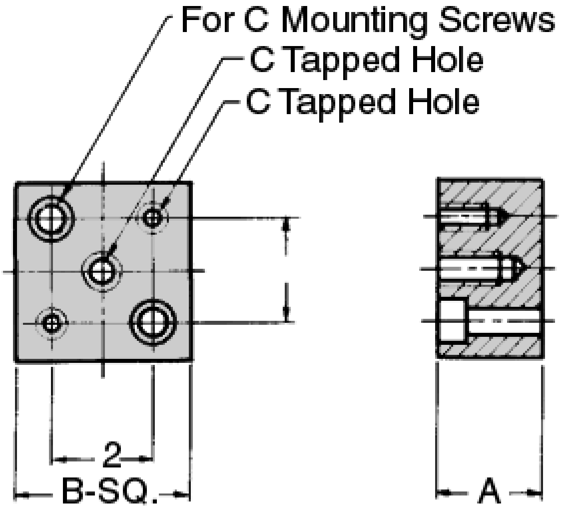


Appendix 8: The Selected Workpiece Specifications



Appendix 9: The Riser Blocks Specifications

MATERIAL: 1041 steel, black oxide finish

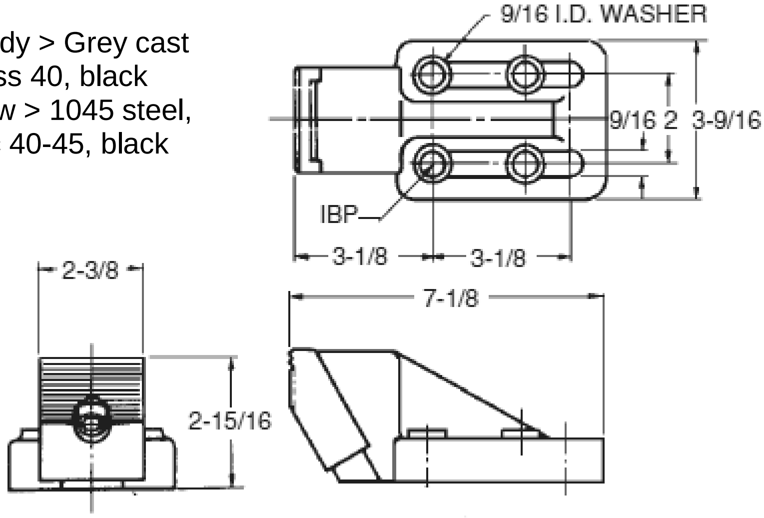


SIZE	PART NO.	A	B	C DIA	SOCKET-HEAD CAP SCREW (2 REQD)
STANDARD	CL-MF40-1601	1"	2-15/16"	1/2"-13	CL-1/2-13x1.50-SHCS
	CL-MF40-1602	2"			CL-1/2-13x2.75-SHCS
HEAVY	CL-MF50-1601	1"	3-3/8"	5/8"-11	CL-5/8-11x2.00-SHCS
	CL-MF50-1602	2"			CL-5/8-11x3.00-SHCS

CarrLane MANUFACTURING CO.

Appendix 10: Downthrust Back Stop Specifications

MATERIAL: Body > Grey cast iron, ASTM class 40, black oxide finish. Jaw > 1045 steel, heat treated Rc 40-45, black oxide finish.

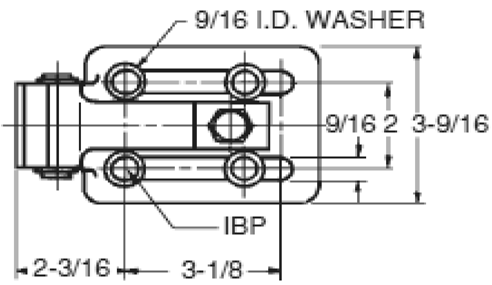


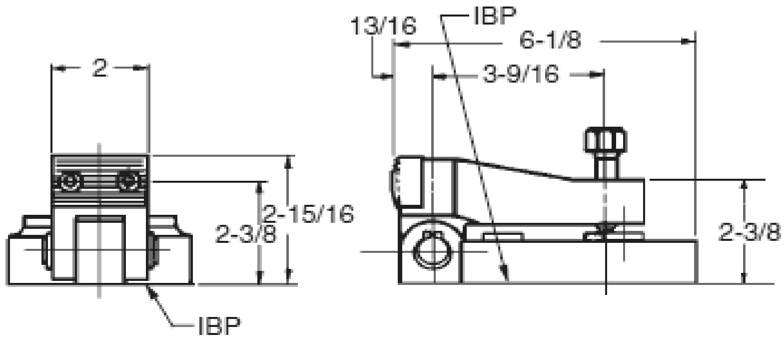
SIZE	PART NO.	SOCKET-HEAD CAP SCREWS (2 OR 4 REQD)	REPLACEMENT JAW
STANDARD	CL-MF40-1101	(4) CL-1/2-13x2.25-SHCS	CL-MF40-1111
HEAVY	CL-MF50-1101	(2) CL-5/8-11x3.00-SHCS	CL-MF40-1111

CarrLane MANUFACTURING CO.

Appendix 11: Pivoting Edge Clamp Specifications

MATERIAL: Body > Grey cast iron, ASTM class 40, black oxide finish. Jaw > 1045 steel, heat treated Rc 40-45, black oxide finish. Clamping screw > High-carbon steel, heat treated, black oxide finish.





SIZE	PART NO.	MAX. CLAMPING FORCE (N)		SOCKET-HEAD CAP SCREWS (2 OR 4 REQD)	REPLACEMENT JAW
		HORIZONTAL	VERTICAL		
STANDARD	CL-MF40-1001	8900	980	(4) CL-1/2-13x2.25-SHCS	CL-MF40-1011
HEAVY	CL-MF50-1001	19000	3200	(2) CL-5/8-11x3.00-SHCS	

CarrLane
MANUFACTURING CO.

Appendix 12: The Developed VB Codes for Adding the Fixture Elements

```
Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swModel = objSolidWorks.ActiveDoc
Dim CompNames(0) As String
Dim vComps As Variant
vComps = swModel.AddComponents((CompNames), (Transforms))
swModel.ViewZoomtofit2
End Sub

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Dim swModel As SldWorks.ModelDoc2
Set swModel = objSolidWorks.ActiveDoc
Dim CompNames(0) As String
```

```
Dim vComps As Variant
vComps = swModel.AddComponents((CompNames), (Transforms))
swModel.ViewZoomtofit2
End Sub

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swModel = objSolidWorks.ActiveDoc
Dim CompNames(0) As String
Dim vComps As Variant
vComps = swModel.AddComponents((CompNames), (Transforms))
swModel.ViewZoomtofit2
End Sub

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swModel = objSolidWorks.ActiveDoc
Dim CompNames(0) As String
Dim Transforms(15) As Double
Transforms(0)=1
```

```
Dim vComps As Variant
vComps = swModel.AddComponents((CompNames), (Transforms))
swModel.ViewZoomtofit2
End Sub

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Sub main()
Set swModel = objSolidWorks.ActiveDoc
Dim CompNames(0) As String
Dim vComps As Variant
vComps = swModel.AddComponents((CompNames), (Transforms))
swModel.RotateComponent
boolstatus = swModel.Extension.SelectByID2("", "FACE", -0.4037392805095, 0.191491914306,
0.3452716403103, False, 0, Nothing, 0)
swModel.ClearSelection2 True
swModel.ViewZoomtofit2
End Sub
```

Appendix 13: The Developed VB Codes for Automating Fixture Elements Assembly

```
Sub main()

Dim boolstatus As Boolean
Dim longstatus As Long
Set objSolidWorks = New SldWorks.SldWorks
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.1491769570417, 0.2389194294356,
0.2222602258129, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.03054176525694, 0.07545823474305,
0.2199692013458, True, 1, Nothing, 0)
Dim myMate As Object
 ("", "FACE", 0.0922756090352, 0.179857158362, 0.2244972606054, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.123321758926, 0.05932175892599,
0.1038717492195, True, 1, Nothing, 0)
Set myMate = swModel.AddMate3(4, 1, False, 0.09399852070358, 0, 0, 0.001, 0.001,
0.5235987755983, 0.5235987755983, 0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
End Sub

Sub main()
Dim boolstatus As Boolean
Dim longstatus As Long
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.17, 0.005273173233036,
0.1540452557745, True, 1, Nothing, 0)
swModel.ClearSelection2 True
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.1702380237501, -0.000184875798368,
0.09706913104702, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.3933062959771, -2.860597430185E-
04, 0.1728420469362, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(5, -1, False, 0.29, 0.29, 0.29, 0.001, 0.001, 0, 0.5235987755983,
0.5235987755983, False, longstatus)
swModel.ClearSelection2 True

swModel.EditRebuild3
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.10168002521, 4.238525435198E-04,
0.2447120408373, True, 1, Nothing, 0)
```

```
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.1560793621731, -4.820820115015E-
04, 0.3932655194037, True, 1, Nothing, 0)
Set myMate = swModel.AddMate3(5, -1, False, 0.29, 0.29, 0.29, 0.001, 0.001, 0, 0.5235987755983,
0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
swModel.ViewZoomtofit2
End Sub

Sub main()
Dim boolstatus As Boolean
Dim longstatus As Long
objSolidWorks.Visible = True
Dim swModel As SldWorks.ModelDoc2
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.01839843730374, 0.2752110322098, -
0.01201256999275, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.04972361978395, 8.729106634178E-
04, 0.3676702442841, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(1, 1, False, 0.001, 0, 0, 0.001, 0.001, 0.5235987755983,
0.5235987755983, 0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
End Sub

Sub main()
Dim boolstatus As Boolean
Dim longstatus As Long
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.1246340732295, 0, 0.3737372818263,
True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.07353852785116, 0.25,
0.3239059690717, True, 1, Nothing, 0)
Set myMate = swModel.AddMate3(0, 1, False, 0.25, 0, 0, 0.001, 0.001, 0, 0.5235987755983,
0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
swModel.ViewZoomtofit2
End Sub

Sub main()
Dim objSolidWorks As SldWorks.SldWorks
Dim boolstatus As Boolean
Dim longstatus As Long
Set objSolidWorks = New SldWorks.SldWorks
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
```

```
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.0631230626239, 0.274745133787, -
0.055454482855, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.1021572791724, -5.732665542837E-
04, -0.1416418810753, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(1, 1, False, 0.001, 0, 0, 0.001, 0.001, 0.5235987755983,
0.5235987755983, 0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
End Sub

Sub main()
Dim boolstatus As Boolean
Dim longstatus As Long
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.2675137701884, 0, -
0.06124025063497, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.05637241961307, 0.25, -
0.1006806557063, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(0, 1, False, 0.25, 0, 0, 0.001, 0.001, 0, 0.5235987755983,
0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
swModel.ViewZoomtofit2
End Sub

Sub main()
Dim boolstatus As Boolean
Dim longstatus As Long
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.1223676941368, 0.275610948425, -
0.04466233385068, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.09542145898138, 0.02510347609086, -
0.1329716473293, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(1, 0, False, 0.001, 0, 0, 0.001, 0.001, 0.5235987755983,
0.5235987755983, 0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
End Sub

Sub main()
Dim objSolidWorks As SldWorks.SldWorks
Dim boolstatus As Boolean
Dim longstatus As Long
Set objSolidWorks = New SldWorks.SldWorks
objSolidWorks.Visible = True
```

```
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.09780215089165, 0.02539999999999, -
0.1234941552973, True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.06505125074682, 0.25, -
0.08787559227693, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(0, 1, False, 0.2246, 0, 0, 0.001, 0.001, 0, 0.5235987755983,
0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
swModel.ViewZoomtofit2
End Sub

Sub main()

Dim boolstatus As Boolean
Dim longstatus As Long
objSolidWorks.Visible = True
Dim swModel As SldWorks.AssemblyDoc
Set swModel = objSolidWorks.ActiveDoc
Dim myModelView As Object
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.00385584136326, 0
Set myModelView = swModel.ActiveView
, -0.04544669051117
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.02313504817956, -0.09846782944087
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.02699088954282, -0.1742123136262
Set myModelView = swModel.ActiveView
= swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.03787224209264
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.06817003576676
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.007711682726521, -0.0530211389297
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.00385584136326, -0.06817003576676
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.00385584136326, -0.0530211389297
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.00385584136326, -0.01514889683706
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.007711682726521, -0.02272334525559
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.0192792068163, -0.09846782944087
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.01514889683706
Set myModelView = swModel.ActiveView
```

```
myModelView.RotateAboutCenter 0, -0.007574448418529
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.03787224209264
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0.00385584136326, -0.01514889683706
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.01514889683706
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.007574448418529
Set myModelView = swModel.ActiveView
myModelView.RotateAboutCenter 0, -0.007574448418529
swModel.ViewZoomTo2 0.0212181493487, 0.1251350780451, 0.1529657559701,
0.1094381314061, 0.0034138369785, 0.1529657559701
boolstatus = swModel.Extension.SelectByID2("", "FACE", 0.1433363482482, 0.1158232641093, 0,
True, 1, Nothing, 0)
boolstatus = swModel.Extension.SelectByID2("", "EDGE", 0.08927657952108, 0.08405116105402, -
0.002495254901532, True, 1, Nothing, 0)
Dim myMate As Object
Set myMate = swModel.AddMate3(0, -1, False, 0.002588842402813, 0, 0, 0.001, 0.001,
0.5235987755983, 0.5235987755983, 0.5235987755983, False, longstatus)
swModel.ClearSelection2 True
swModel.EditRebuild3
swModel.ViewOrientationUndo
swModel.ShowNamedView2 "*Trimetric", 8
swModel.ViewZoomtofit2
End Sub
```