2016-03-01

Decomposition of Manufacturing Processes for Multi-User Tool Path Planning

Andrew Scherbel Priddis
Brigham Young University

Decomposition of Manufacturing Processes for Multi-User Tool Path Planning

Andrew Scherbel Priddis

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Walter E. Red, Chair
C. Greg Jensen
Michael Miles

Department of Mechanical Engineering

Brigham Young University

March 2016

ABSTRACT

Decomposition of Manufacturing Processes for Multi-User Tool Path Planning

Andrew Scherbel Priddis
Department of Mechanical Engineering, BYU
Master of Science

Engineering activities by nature are collaborative endeavors. Single-user applications like CAD, CAE, and CAM force a strictly serial design process, which ultimately lengthens time to market. New multi-user applications such as NXConnect address the issue during the design stage of the product development process by enabling users to work in parallel. Multi-user collaborative tool path planning software addresses the same serial limitations in tool path planning, thereby decreasing cost and increasing the quality of manufacturing processes. As part complexity increases, lead times are magnified by serial workflows. Multi-user tool path planning can shorten the process planning time. But, to be effective, it must be possible to intelligently decompose the manufacturing sequence and distribute path planning assignments among several users. A new method of process decomposition is developed and described in this research. A multi-user CAM (MUCAM) prototype was developed to test the method. The decomposition process and MUCAM prototype together were used to manufacture a part to verify the method.

Keywords: decomposition, multi-user, CAM, tool path planning

ACKNOWLEDGMENTS

First and foremost I would like to acknowledge the help and support of my wonderful wife. She has been unwavering in her support of my research as well as an invaluable resource with whom I can discuss and further develop ideas.

I would also like to acknowledge the help and direction given by Dr. Red, who was my advisor and mentor. His guidance and direction helped to shape and focus the research. More importantly he pushed me to take this research further and helped me to accomplish more than I thought was possible.

Also it is important to acknowledge the researchers who have come before me in the development of multi-user CAx applications. Their work provided an environment where a prototype could be developed and tested without creating everything from the ground up. Also, the insights provided by their work gave direction to my work to further the body of research.

This work has been performed under the NSF Center for e-Design, and with the assistance of industrial members of BYU's research site.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

T_{SU}	Total time required to generate tool paths for a part by a single user
t_{CAM}	Time required to plan the toolpaths
t_{decomp}	The time spent decomposing a part into a series of operations
t_{setup}	Time needed to setup CAM environment
t_{finish}	Time needed perform finishing tasks such as post processing and verification
$T_{MU,elapsed}$	Total elapsed time required to generate tool paths for a part by a multi-user team
N_{users}	Number of users working together on a multi-user CAM team
e	Efficiency of multi-user team compared to single user
$T_{MU,manhours}$	Total man hours required to generate tool paths
$\Delta T_{elapsed}$	Elapsed time saved using multi-user CAM vs single-user CAM
$\Delta T_{manhours}$	Man hours saved using multi-user CAM vs single-user CAM

CHAPTER 1. INTRODUCTION

Single-user tools constrain developers in collaborative work environments at every step of the product development process, from idea generation to manufacture. Red et al. wrote that "a dichotomy exists in the engineering design process between 1) product teams organized to engineer products collaboratively, and 2) the single-user architectures inherent in computers and computer-aided design applications (CAx)." Further, "single-user serial architectures inhibit concurrent engineering, in spite of the numerous research efforts into product team cooperation, functional constraints, and data/model propagation and transparency." [3] Modern CAx applications, like CAD, CAE, and CAM, fall into this category, forcing a strictly serial design process, which ultimately lengthens time to market. Also, as part complexity increases, lead times are magnified by serial workflows. These serial processes are contrary to collaborative principles that development teams strive to apply to product development.

New multi-user CAx applications such as NXConnect [4] are designed based on collaborative principles and enable users to work in parallel during the design stage of product development, thus bringing products to market more quickly. Many researchers have recognized the need for a multi-user environment for product development. Hepworth [5] lists ten different multi-user CAD systems that have been developed to increase collaboration. Multi-user tool path planning, or mutli-user CAM (MUCAM), is being developed as a software that would provide a similar environment for parallel work during the manufacturing stage of product development. The developers of NXConnect have shown that "collaboration decreases the product development time in proportion to the number of multi-users." [6] By creating a collaborative environment for manufacturing tool path planning, a similar improvement in efficiency is expected, enabling companies to bring products to market more quickly.

Marshall stated, "For any such collaborative environment to function effectively, there has to exist rules of interaction to govern the work of the multiple users, preventing or resolving conflicts between them." [7] Marshall, as well as other researchers, developed methods to to facilitate work within a multi-user CAD environment. For MUCAM to be successful a method must be developed specifically for generating tool paths in a collaborative setting.

1.1 Problem Statement

One might question if it possible for several users to simultaneously generate tool paths for a single part in a parallel workflow. The aim of this thesis is to answer this question and identify principles and processes that can be used to decompose a manufacturing process into a series of steps that can be entered into CAM software simultaneously by multiple users. As part of this research, a prototype of MUCAM software was developed to provide a multi-user collaborative tool path planning environment to verify the process decomposition method.

1.2 Research Objectives

This research focuses on the following specific objectives which serve to answer the questions in the Problem Statement:

- Develop and program Multi-user CAM functionality into NXConnect.
- Develop a method to distribute operationally independent feature groups among team members.
- Use Multi-user CAM to verify the method of distribution among users.
- Use software to plan the tool paths for a part and manufacture it as a final proof of concept.

CHAPTER 2. BACKGROUND

Mutli-user CAM is built upon previous software development and research. This includes the development of numerous single-user CAD and CAM applications as well as several multi-user applications. These are important to discuss as they have all influenced the development of the MUCAM prototype. Each one will described and discussed in the following sections to aid the reader in understanding how MUCAM works and how it differs from single-user CAM.

2.1 Tool Path Planning: Single User CAM

2.1.1 Overview

Multi-user tool path planning is completed using a process similar to the process used in typical single user tool path planning. The process variations used in MUCAM will be described in Chapter 4. Details of the single user process are taken from the NX CAM training manual. [8]

Modern large scale CAD packages, like the one shown in Figure 2.1, generally integrate manufacturing tools that assist in tool path planning. Third party applications are also available. At the beginning of the planning stage, a solid CAD model is opened in the CAM application. The application is then used for planning setups, defining parameters and tools, creating CNC tool paths, and so on. In single user applications the planning processes for both simple and extremely complex parts are planned serially, feature-by-feature.

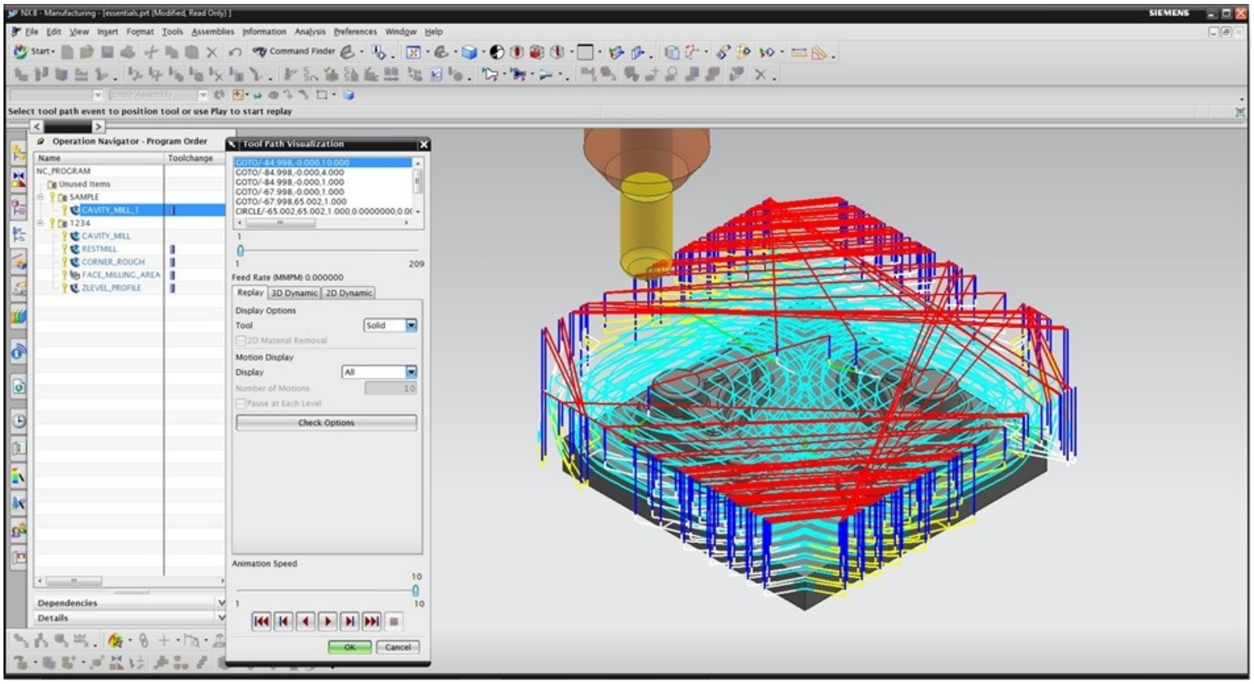


Figure 2.1: Tool path verification using Siemens NX

A single user may not be knowledgeable in all relevant processes, such as tool selection, cut pattern selection, matching machine to process features, and also considering whether equipment is available for all planned operations. In single-user CAD, designers rarely have sufficient information to complete complex designs independently without some external help and tutoring. Similarly, tool path planners require away-from-desk inquiry to gain sufficient know-how to complete the process planning.

Figure 2.2 [8], also from the NX CAM training manual, shows the steps of the process used to develop tool paths by a single user. The user must setup the CAM environment, generate geometry, define CAM objects and operations, generate paths, finalize the process, and so on. These steps will be described in more detail in the following paragraphs. Figure 2.2 also shows that the definition of parent groups (Program, Tools, Geometry, Method) can be completed in any order. However, because only one user can use the application at a time the work proceeds in a serial fashion. The process must cycle after each operation is created until every required operation has been defined. It will be shown later that multi-user CAM differs by removing the unnecessary process cycles. It is also important to note that the single user process has no specific step for the decomposition of the part into a series of operations. As explained later in more detail, the single user is free to decompose as needed whereas multi-users require a specific decomposition plan before proceeding in the process.

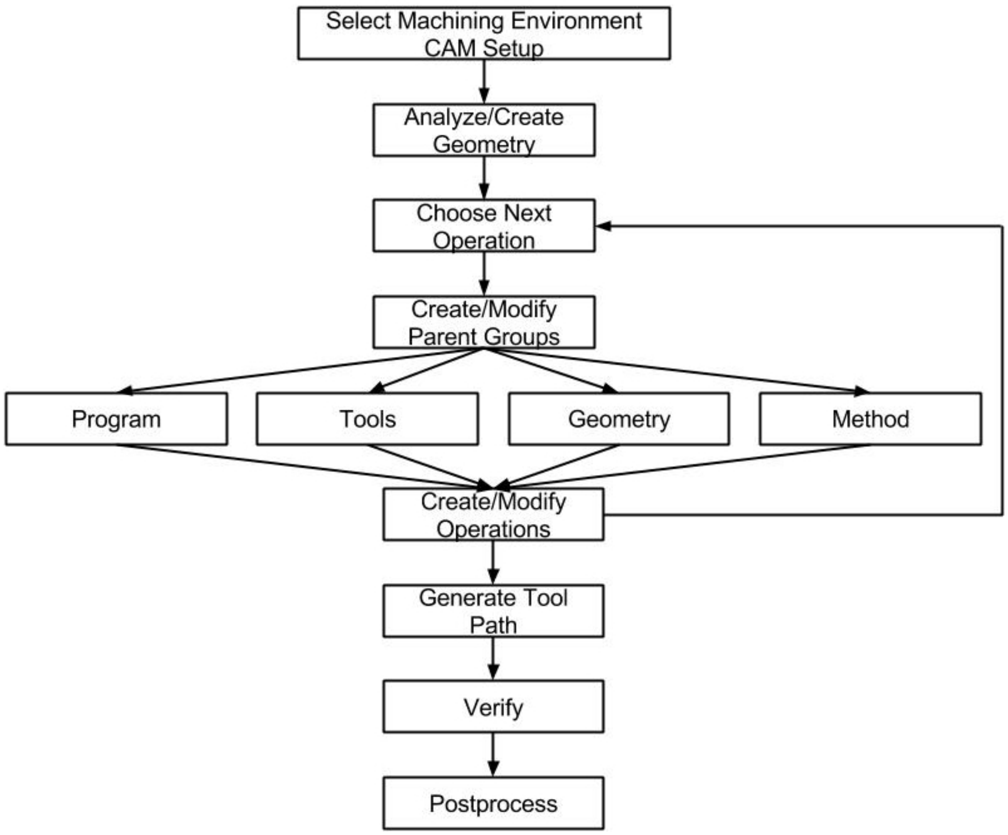


Figure 2.2: Single user tool path planning loops until every operation has been defined

2.1.2 Process steps

Once the part model design is finished, the process of tool path development begins. In each CAM application the first step is to complete the initial setup of the part within the application. This setup creates the proper environment to develop tool paths. Many machining processes are supported by CAM applications and so selecting the proper environment ensures the proper tools and operations are available to the user. For example, turning operations would not be needed by a user creating 3-axis milling operations, so the milling environment is selected and then only milling operations are available to the user in the CAM software GUI (Graphical User Interface).

After the initial setup is completed, the next step is to create any supporting geometry. This includes defining the stock material size and shape, setting the machine coordinate system, specifying the machine to be used, modeling mounting mechanisms, and identifying the work piece. These steps ensure that the tool paths are generated based on a correct representation of the physical environment and material.

Tool path planning is defined by a planned sequence of operations. A single user approaches the creation of operations by selecting an appropriate cutting pattern, tool, depth per cut, cut area and so forth. Any need to change any of these parameters means that a new operation must be created. When creating operations, the paths are generally created in the order that the machine will perform them. The user will generally start by generating a roughing operation, then analyze the remaining material, decide on the next best operation, analyze the remaining material, select the next best operation and continue until all the excess material has been addressed. This is a logical and effective method for defining all tool paths. An important feature of CAM applications is that they do not require operations to be created in a particular order. The operations can be created and defined in any order and then can be reordered to the correct sequence before finalizing and generating the CNC code to avoid breaking tools or machines.

Before an operation is generated, the user needs to make several decisions. These include which tool to use, which areas to machine, what method (roughing, semi roughing, etc.), and into which program the operation should be placed. These decisions are made by creating tool, method, geometry, and program objects in the application that are then referenced when an operation is defined. It is possible to create all these objects initially and then create all the operations, or to focus on a specific operation and create the needed objects for that operation just before creating it.

Once planned, the process operations are conducted by the appropriate software operation by operation. As an example, the processes for developing the tool paths for an engine block are shown in Figure 2.3. First, the stock piece is defined. Next, the roughing operations are defined and planned, then semi-roughing, followed by the finishing operations. Last any final features such as slots, chamfers and holes are added.

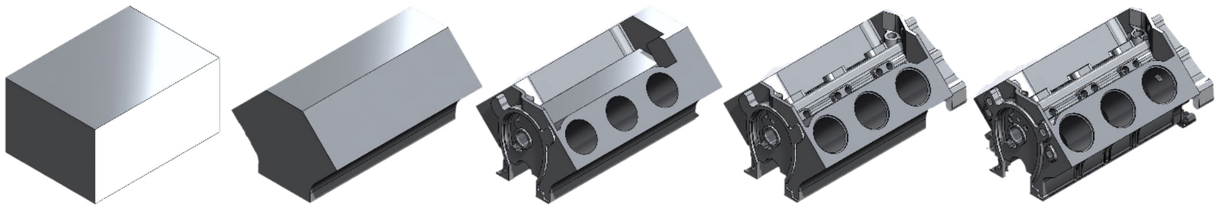


Figure 2.3: Typical serial progression of operations for an engine block

Once all the operations have been defined the entire set of operations is verified to ensure accuracy and then post-processed and exported as machine-readable code. The machine readable code is exported as a program containing all the instructions needed to perform a set of operations in a manufacturing process. If a part is to be moved to another machine, another program would be exported containing those instructions.

2.1.3 Review of single user CAM

Many researchers have expressed concerns with this process. Speaking of traditional single user software, Okulicz stated,"process planning represents a very weak link within the product realization process and is a major source of inefficiency, errors and duplicative steps." [9] This idea was reiterated by Dong: "There is an invisible wall between design personnel and manufacturing personnel in this traditional design and manufacturing environment. This wall blocks the information flow between design and manufacturing, and greatly increases product development time and cost." [10] Further, Yau found that,"strong interdependency among design, manufacturing, and inspection are relatively unexplored or simply neglected. As a result, low efficiency and high production reduce the competitive edge of the industry. Therefore, there is an ongoing trend of developing concurrent engineering techniques." [11]

The problems these researchers describe arise because, as Red states "Concurrent engineering is limited by the serial design of modern CAx tools like CAD/CAE/CAM; these tools limit model and assembly creation and editing to one active user/engineer." [1] In a collaborative design process this creates a significant bottleneck in the overall design process. Multi-user CAM has the potential to remove these walls by providing an environment where design and manufacturing personnel have access to the same part files at the same time, encouraging collaboration among engineers as the parts take their final shape.

2.2 Multi-User Applications

2.2.1 CoWord and CoPowerpoint

Many researchers have recognized the need for concurrent software tools. Several of these researchers have developed software to explore the potential of multi-user applications. Sun et al. successfully created created multi-user plugins for single-user software. Specifically they created CoWord and CoPowerpoint. [12] These applications were created by using the program Application Programming Interface (API) to apply a multi-user framework to the software. Using the API allowed them to achieve the design objectives they had identified at the outset of the project:

- Retain original file formats
- Retain original features and functions
- No change should be made to the original source code
- The application response time should not be slowed
- Users should be free to perform any operation at any time
- Users should know with whom they are working
- Users should know what other users are doing.

The successful implementation of the project objectives is significant because these application features have propagated into most multi-user applications.

2.2.2 Google Office Applications

Similar to CoWord and CoPowerpoint are the Google suite of office applications. These differ slightly from other multi-user application in that they were designed from the start to support multi-user simultaneous editing whereas other multi-user software is built around existing software. Google allows up to 50 users to "edit or comment on a document, spreadsheet, presentation, or drawing at the same time." [13] The Google office applications are based on a thick server with weak clients. The clients in this case are considered weak because the application runs completely within the architecture of an internet browser. Also notably, changes made by the users are stored in an online database and then pushed to all other users. Because the Google applications were built from the ground up, the first four objectives Sun et al. enumerated are not applicable, but the other three objectives are a part of the software.

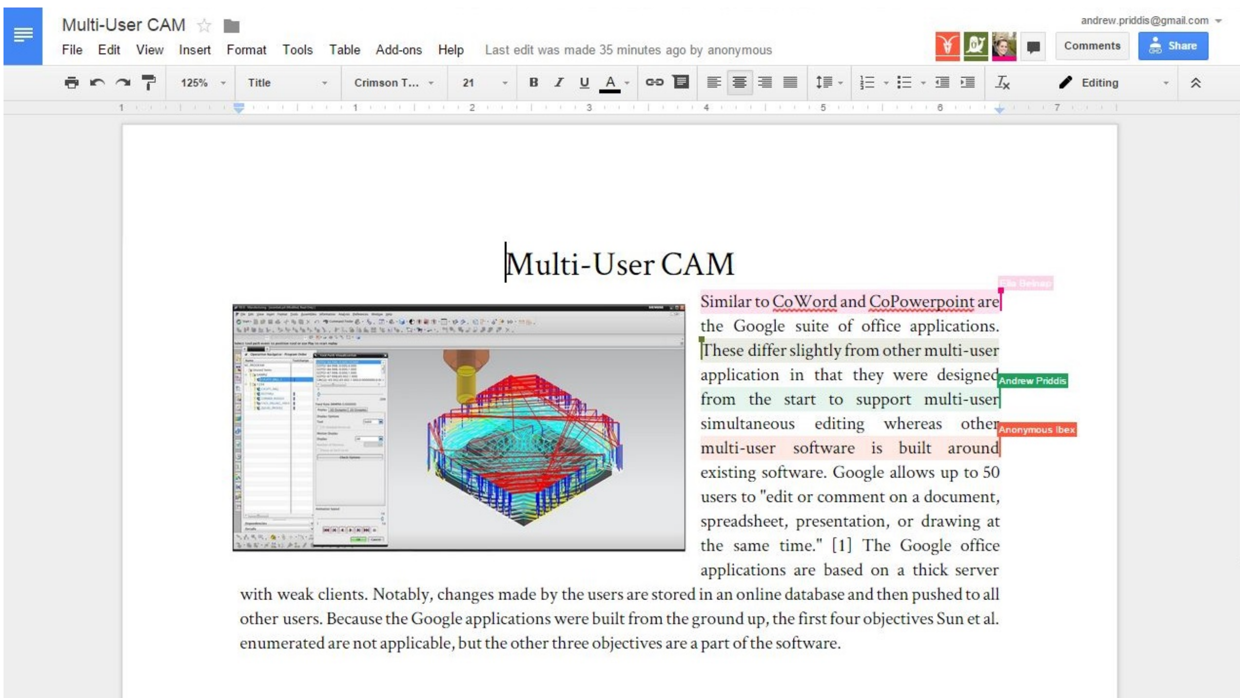


Figure 2.4: Several users editing document using Google docs

2.2.3 Multi-player Gaming

Long before research began into multi-user CAX tools, multi-player video games were developed. Multi-user CAx tools have strong similarities to multi-player gaming, thus gaming has been used by researchers to show the feasibility of multi-user CAx software. French et al. at length discussed the similarities between Minecraft, a popular multi-player game, and mutli-user CAx applications. They explored how teams of users together approached the design and modeling of highly complex models within the Minecraft environment. More importantly they used information about how players design and build within Minecraft to provide insight into how engineers can "effectively build, communicate, and manage projects." [14]

Environments such as multi-player gaming, especially games in which the players design and build together, lend credence to the idea that multi-user CAx tools can be successfully implemented. Marshall identified specific principles that are used in gaming that can be applied to a collaborative engineering environment [7]:

- Constrain users by experience level.
- Give users specific tasks to complete.
- Provide a means for users to communicate with one another and see the work of others.
- Allow users of different skill level and skill-set to work together towards a common goal.

2.2.4 Other Types of Collaboration

Research on different types of collaborative engineering applications has been conducted by several groups of researchers. Ming et al. [15] focused on collaboration between Computer Aided Process Planning software (CAPP) and Computer Aided Manufacturing (CAM) software. The software they created allowed data to be passed between CAM and CAPP software. They found that in the manufacturing portion of the collaboration there were several benefits. By allowing the different applications to collaborate they were able to reduce rework in parameter selection, more efficiently generate NC programs, and optimize the sequence of operations to be used in part manufacture.

2.2.5 NXConnect

A considerable amount of research has been done at BYU to develop the multi-user CAD application known as NXConnect. NXConnect is a multi-user CAD application that lets several users access a part file simultaneously and model the part while changes made by a user are updated in real time to other users. As this data is passed from one client to another it is also stored in a database so that it is available for access later. [3] In a 2011 paper, Xu describes the architecture and implementation of NXConnect, which is the platform for the multi-user CAM environment.

Notably Xu mentions that NXConnect is based on a thin server, with strong clients. Each user has Siemens NX installed locally and data is passed from the server to the client and back. Since the local stations (clients) perform the computations, a real time multi-user experience is delivered. [4]

Much of the research conducted using NXConnect is especially applicable to decomposition for multi-user tool path planning. Marshall used NXConnect to research how to decompose a part into a set of features and regions that can be modeled simultaneously by multi users. [7] Within multi-user CAD, part decomposition is more complex than in multi-user CAM because part features are based on previously created features. Changes to features higher up the part tree can affect all of the following features. Marshall created a method that would analyze the features for dependencies then make assignments to users that would result in the fewest conflicts. Thus each user could work in the assigned region with little concern for what other users are doing. Multi-user CAM is unique in that operations can be created any order as long as they are reordered before post-processing. This means that decomposition methods will differ from those previously developed for multi-user CAD.

2.3 Part Decomposition

Research related to part decomposition has been conducted by several groups of researchers. Sarma et al. researched how to minimize the number of setups and tool changes. They wrote, "Planning is a complex task. Traditionally, this complexity has been contained by decomposing higher level tasks into lower level subtasks, which can be handled independently. Accordingly, manufacturing features are broken down into operations." [16] Sarma et al. points out that in the past a large body of research was developed in the areas of features, feature recognition and manufacturing features. [17–36] Once a part has been decomposed into independent operations, process planning becomes a manageable task of generating tool paths for individual operations.

Wei et al. focused their efforts on an automated decomposition method that would generate all possible methods of manufacturing a part. A user can then analyze the various methods and identify the best method for the given situation. [37] Others such as Yahia et al. [38] have followed Wei by creating means to optimize the sequence of operations to most efficiently manufacture a part. The tool paths must still be generated by users, but it is important to note that the manufacturing process can be identified at the beginning of the manufacturing planning process using only the completed part model.

Rho et al. [39] built upon the work of Wei by identifying thirty six features which they term ”atomic features’ These are the features that are most often machined into a part. Kim et al. [40] used the thirty six atomic features to create a method to automatically recognize these features and identify precedence relationships. Once the relationships are identified, it is easy to see how different operations can be divided up among different users for tool path planning. These findings by Rho and Kim are significant because they demonstrate the principal of feature/process independence. Atomic features can be considered independently by various users and defined independently, even if a feature has a precedent relationship with another.

Marshall [7] investigated a general decomposition method for design in a collaborative CAx environment. Marshall's work focused on how to divide the design space while modeling in a CAD application, but the work is very applicable to model decomposition for CAM. Marshall argued that “For any such collaborative environment to function effectively, there has to exist rules of interaction to govern the work of the multiple users, preventing or resolving conflicts between them.”

Marshall described the principles that the rules are based on. With respect to interacting with other users work she stated: “The coordination method for constraining users should not completely isolate users from each other, thereby eliminating collaboration, but should allow them to see the developing work of others, and encourage their communication.” In a CAM environment, expertise in manufacturing processes is invaluable and so experts would ideally be used on certain tasks. This idea meshes well with Marshalls findings: “The coordination method should distinguish between the roles of users, giving each user ownership of a portion of the design space by assigning tasks to those most qualified...”

In addition to these principles, Marshall listed a final extensive list of design requirements for a collaborative CAD environment that can be adapted to a CAM environment. In studying the literature Marshall found that the users should not be completely isolated from each other, should be able to see each others work, have a specific role and task, be able to communicate with each other, and be able to maintain a continuity in the work space.

Marshall also reviewed existing collaborative environments and found that the application should be able to constrain users by skill level or team function, allow for a cross-functional team to work together simultaneously, allow a single decision-maker to govern conflicts, and allow an authority figure to assign tasks.

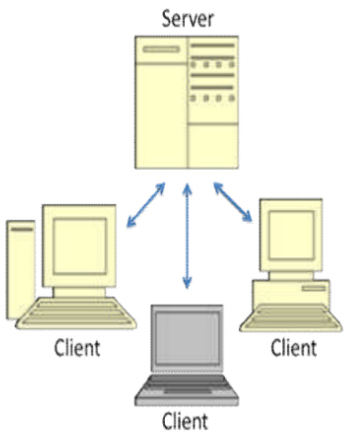
CHAPTER 3. MULTI-USER TOOL PATH PLANNING PROTOTYPE

3.1 Overview

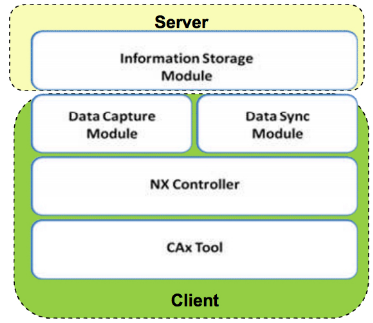
The Multi-user CAM software that was created differs greatly from single user CAM because several users can engage the model and plan tool paths at the same time, as shown in Figure 3.1.

Figure 3.1: Mutli-user CAM prototype demonstration

The first multi-user tool path planning prototypes are designed to run on the NXConnect architecture. [4] This was done because NXConnect research demonstrated that the NX API (application programming interface) could support a multi-user environment. Further research demonstrated that this ability extended to tool path planning functions. Not all CAM applications have



(a) CS architecture



(b) NXConnect modules

Figure 3.2: NXConnect architecture [1]

an open interface to allow a multi-user environment. For example, research into PTC Creo has shown that much of the manufacturing functions are not available in the API. This could also be true for other systems. The NX API is very well suited to multi-user tool path planning; thus, an application can be written that allows multiple users to simultaneously define tools, operations, and supporting geometry.

3.2 Architecture

The MUCAM prototype is based upon the NXConnect Architecture described by Red et al. As shown in Figure 3.2(a), like NXConnect, the MUCAM application "utilizes Client-Server (CS) with a thin server and strong client. The server stores the data for the part file and broadcasts changes to each client workstation. Each client maintains a local copy of the part file which is constantly updated." [1] The local copy of the part file must be run in a full version of the CAM software and for this reason the client is a strong client.

The MUCAM prototype was developed as a plugin to Siemens NX. The strong client allows each user to run a full instance of Siemens NX. The user then has access to all the capabilities built into NX as well as a familiar environment in which to work. The thin server acts as a data storage device as well as a message passing module.

3.2.1 Client

When the MUCAM application is loaded, the modules shown in Figure 3.2(b) are enabled. The Data Capture Module monitors the users actions waiting for a change to be committed to the CAM setup. These changes include the creation, edit, or deletion of operations, tools, geometry, methods, or programs. When a commit is detected, the commit is captured by the Data Capture Module and all of the data is extracted. This data includes the type of object, a creator ID for each object created, the part it belongs to, as well as all parameters that define the object. This data packet includes everything needed to recreate the object on another client. The data packet is translated into a single message by the NX Controller and is sent off to the server to be stored as well as passed to the other clients.

On each client the Data Sync Module is listening for messages that have been passed along by the server. When the application receives a message it is translated back into usable data: object type, who created it, the part is belongs to, and object parameters. An object of that type is created and the object parameters are used to update the parameters of the new object on the client. The object is then committed to NX where it appears in the users object tree.

3.2.2 Server

The server first acts as a connection point for all the clients. The server accepts connections from clients, registers the users and then becomes the common point by which the clients can communicate with one another. After the users have connected, the server has two main tasks: first, pass messages between clients and second, store all the data committed in a CAM session. These tasks are relatively simple and so little work is actually done by the server. The heavy lifting is done at the client level. This includes geometry visualization, message generation, message reception and application, and the actual generation of tool paths.

When the server receives a message it performs certain actions based on the message type. When a new user joins the session, the server receives a Connection message. The server registers the client so messages can be exchanged. The server also sends that user a list of the parts saved in the database. When a user wants to open a part, a Part Change message is received and the server sends the user a copy of the most recent version of the part which is opened by the user. Database

Messages contain data of commits within the CAM session. The data in these messages is stored in a table in a database for that part. These messages are also forwarded to all users connected to that session.

Data from database messages is received as a single string and is entered as a single line in the database table. These messages can hold hundreds of parameters in a single message. The long strings of parameters are generated on the client by passing a CAM object into a serialization library. The entire table of parameters is converted into a single string of parameters and values. The string is then sent to the server as a single message. When a client receives a message from the server it is passed back into the serialization library and the sting of parameters is returned to a table of parameters and values and applied to a new CAM object. By storing the parameters in a single string, complicated logic and gigantic database tables are not needed. This simplifies the work performed by the server, simplifies implementation of the server, and once again allows the heavy work of message interpretation and implementation to be performed by the clients.

3.3 Implementation

The NX API does not allow access to the NX event handler. The only way to monitor activity is to use callbacks. These callbacks allow the application to call custom functions when icons are clicked within NX. More specifically, callbacks are used to call custom functions at the conclusion of the original function called by NX. For example when a user clicks the operation creation icon, at the completion of the function, the MUCAM operation function is called to capture the data and pass it to the other clients.

The NX API generally allows the user to access most of the data in the models. However, NX does not currently allow the CAM object type and subtype parameters, which are needed for any object creation, to be returned through the API. To capture this data, an extra dialog box is generated to ask the user to enter the type and subtype manually. The dialog box looks identical to the NX dialog box that first appears when an operation is created. The user is required to click on the same icon that was selected earlier and the data is then captured. Once the type and subtype are known, the objects can be regenerated on any other client. This is the only difference in the way the user interacts with the GUI than with the standard NX GUI. The user will notice no other changes in the method used to interact with the application.

CHAPTER 4. PROPOSED PROCESS DECOMPOSITION METHOD

4.1 Overview

Although multi-user tool path planning is built on the NXConnect architecture and a natively single user application, there are significant differences in how the work progresses from either application. A new strategy, shown in figure 4.1, had to be developed to generate tool paths in a multi-user setting. Existing CAx software does not anticipate multi-user engagement as it only permits one user to interact with a model at a time.

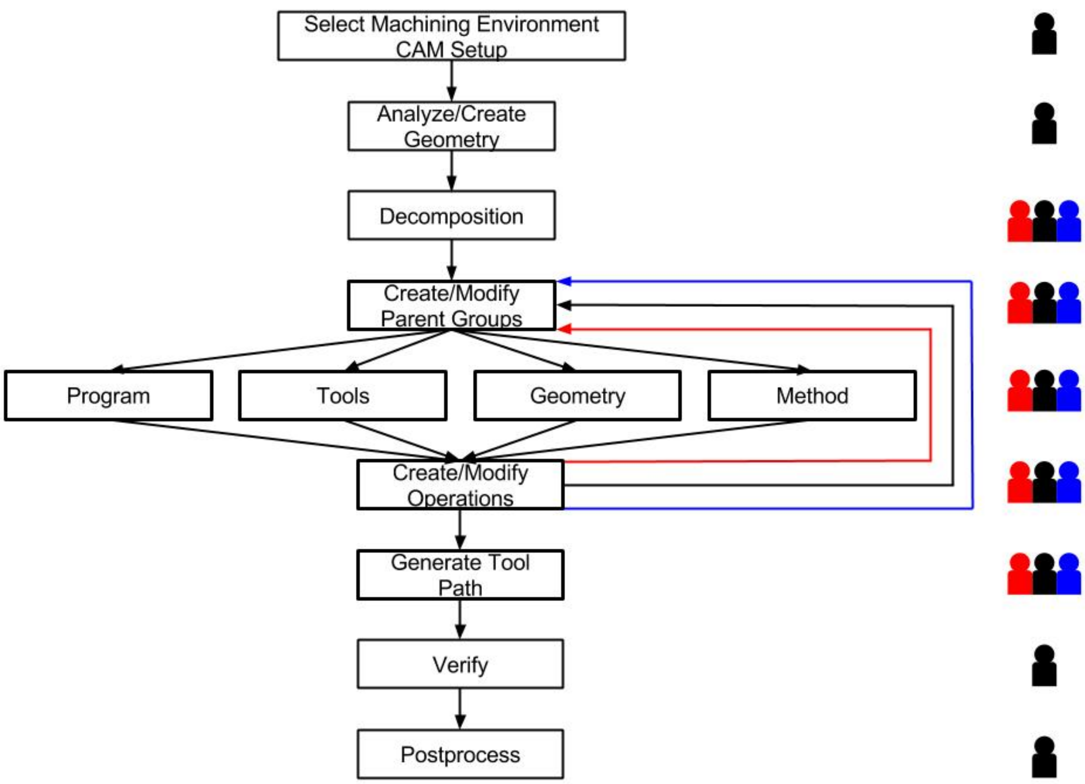


Figure 4.1: Decomposition of part into machining operations

The reader should note that the process shown in figure 4.1 is similar to the process shown in figure 2.2 but has been adjusted to reflect the changes in the tool path planning process required by a multi-user approach. The reader will note that the main differences between figure 2.2 and figure 4.1 are the multiple repetitive cycles (one per user), the new decomposition step to replace operation selection, as well as the addition of the symbols to the right of the figure. These symbols show which steps can be worked on in parallel by multiple users and which must be completed by a single user. The corresponding loops have been adjusted to show the steps each user will repeat individually until all assigned tasks have been completed.

The initial goal was to complete the entire process using a parallel workflow, but it was quickly understood that not all steps require or even allow a multi-user approach. The reason that four steps must be completed by a single user are briefly explained in the following four points respectively:

- The environment and CAM setup is a short step but must be completed before any other work can be started. The parameters set in this step are common on each client computer. A single user sets the parameters and they are passed to the other users.
- The supporting geometry is also common on each client computer and also must be completed before moving forward in the process. This includes defining the work envelope, defining machine geometry, etc. This is also completed by a single-user within the software then sent to the other users.
- The verification step requires a user to simulate the whole process using the operations generated by each user. As there is no benefit to multiple users performing a verification, it is completed by a single user.
- Postprocessing generates code to load onto a CNC machine from the operations created by other users and also only makes sense as a single-user process.

The entire process and reasoning used to develop the process is explained extensively in the following sections.

4.2 Principles of Multi-User CAM

As the MUCAM was being investigated, general principles were identified and then combined to create the MUCAM Method. These principles are enumerated here:

4.2.1 Part Ownership

Each part should be assigned a part manager. This is important so that there is a person responsible for completing the part in a timely manner. Initially the part manager will analyze the part to determine if it is a candidate for MUCAM or if it is better suited for a standard single-user approach.

In the event that the part is determined to be a suitable candidate for MUCAM, the manager selects the team to complete the work based on expertise and availability. This user is also responsible for making sure all single-user steps are completed at the appropriate time in the MUCAM process.

If the part is determined to be only suitable for a single-user approach, the part manager will assign a user or complete the tool path planning.

4.2.2 Part Complexity

As mentioned, not all parts are suitable for path planning by a multi-user team. Often parts require only a handful of operations to be generated. MUCAM is only beneficial if the number of operations is high enough that several users would be engaged continuously for several hours. This principle became obvious while trying to select a part to use for testing by a multi-user teams. Research on this specific principle would be required to give exact criteria. There are several potential indicators that a MUCAM approach would be appropriate listed below but these are suggestions from observations that would need to be researched fully:

- Parts that require multiple machines to complete
- Parts with several contoured faces
- Parts that require multiple setups

- Parts that have a large number of features
- Parts that require several tool changes

4.2.3 Feature Independence

MUCAM is possible because CAM software allows operations to be defined in any order. Thus, the part should be divided into groups of independent features that can then be freely pursued by users simultaneously. They can be divided in whatever way seems fitting to the team. For example, they could be divided by roughing, semi-roughing, finishing, etc.

4.2.4 Machine Independence

Parts that require multiple machines require a separate program to be loaded on each machine. Each program can then be treated as a miniature MUCAM process, or can act as the division point to make assignments for separate users.

4.2.5 Operation Sequence

Before exporting the final code, the operations generated by the users must be rearranged into the order in which they will actually be machined, the order from blank to finished product (roughing to finishing). This is typically overseen by the part manager. The machine code then can be generated and loaded onto the machine(s).

4.2.6 User Expertise

MUCAM allows operations to be defined by users with expertise in that specific machining process. Users with expertise in certain processes should be assigned those sections of the part. This speeds the MUCAM process and leads to more efficient machining.

4.2.7 Training

MUCAM provides a unique training environment in which expert users can train novice users. Novice users should be included in MUCAM teams and allowed to fully interact with the part within the CAM environment while receiving direction and insight from the expert users.

4.3 Multi-User CAM Method

The principles described in section 4.2 were used to develop the method described in the following sections.

4.3.1 Machine Decomposition

Multi-user applications require the users to look ahead at what needs to be done and then create a plan to achieve the final goal. Multi-user CAM requires planning on several levels to be successful. Before planning begins, the part is assigned a part manager (MUCAM principle 1). This is important so that there is a person responsible for the part, ensuring the part is completed, and completed in a timely manner. The part manager then must determine if the part is a candidate for MUCAM or single-user CAM (MUCAM Principle 2). If MUCAM is chosen, then the first step of the planning process is to plan which machine(s) to use. This can be completed by the part manager alone, or the manager can gather the team at this point and employ them to identify the machines needed (MUCAM principle 4).

In the event that a part requires multiple machines or setups, multiple programs must be written. Each program represents a manufacturing process and consists of all the operations completed by a single machine using a particular setup. The users must identify what the in-process work piece (IPW) will be for the beginning of each process. The original stock is the IPW for the first process. A duplicate of the CAD model is made and adjusted to represent the state of the part at the beginning of the next process and so on until there is a model of each part state for the beginning of each process. Each of these parts can be treated as a single machine part and the tool paths can be generated as follows as they are for single machine parts. Therefore, the rest of the MUCAM discussion will address single machine parts.

4.3.2 Part Setup

The setup within MUCAM software is the same as the setup for a single-user system. Initially a CAD part is imported into the CAM software. As discussed for single user CAM, the user must choose the proper environment in which to work. Next the user must define the stock piece, the touch off point, the part orientation and other such tasks. A single user must complete this initial work before other users join the part session because the information generated in the setup is needed by each client to allow the other users to begin other tasks. This ensures that when other users join the session and begin defining operations and tools that the data used by the system is accurate and identical for all users.

The part manager is responsible for beginning the process as well as the finishing steps. The manager initiates the process by selecting the environment, defining the work piece and stock, setting up the touch off point, etc. At this point the part is ready for operation definition by multiple users.

4.3.3 Decomposition of Operations

Although the part is ready for operation definition, users are not yet ready to define the operations. Before the users begin defining the process operations, they must be assigned which operations to work on. This ensures that there will be no overlapping work and that users will work in the most effective manner. Whereas a single user can identify the next appropriate operation after the completion of the previous one, in a multi-user setting the operations must all be identified ahead of time and intelligently divided among the users (MUCAM principle 3).

NC machining and CAM software are unique in that there are natural decomposition methods. Each part is machined in a series of operations that are separated from each other in the software by machining technique, tool used, depth of cut and so on. Generally any time any parameter needs to be adjusted, a new operation is created within the CAM software to represent that movement. For example all of the roughing passes used to remove large amounts of material at the beginning of the process could be a single operation. However, if a second size of mill were needed for roughing, that would be defined as another operation. Each of these operation can be created independently of each other in any order. After the initial setup, the users together decompose the entire process into a set of steps or operations which can each be defined independently of the operations. Each operation can then be assigned to a user to create and define within the CAM software.

There are different methods can be employed to decompose the part. Two methods are described in the following sections.

Machining Order Decomposition

The first approach is similar to the approach that is employed by a single user. The team together will create a list by stepping through the machining process, trying to identify the operations in the order they will actually be performed by the machine(s). The part manager or the whole team together must first identify the setups, then the operations for each setup can be identified. This method was used to decompose the engine block shown in Figure 4.2.

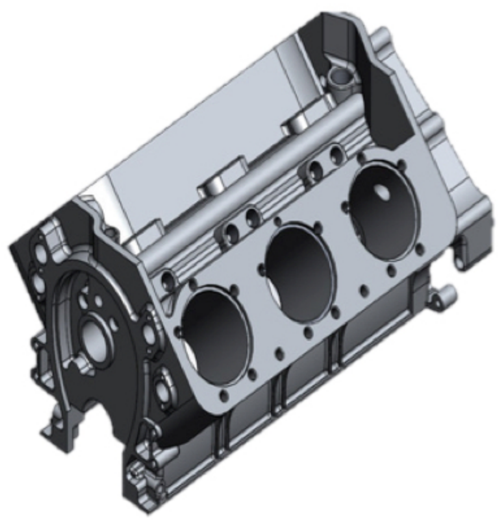


Figure 4.2: Engine block to be machined from Solid Billet [2]

The decomposition of the engine block is enumerated in Table 4.1. The users decided on using two separate machines. Two setups were identified to be completed on the first machine to finish the front and back faces. A third setup on a second machine was identified to machine the other faces.

Table 4.1: Operations for engine block shown in Figure 4.2

Operations		
MOUNT UPRIGHT	**MOUNT END DOWN**	**MOUNT IN 4 AXIS MILL**
Face Front Face of Block	Face Back Face of Block	Face Mill Slanted Faces
Rough Mill Front Face	Rough Mill Back Face	Face Mill Top Face
Semi Rough Front Face	Semi Rough Back Face	Rough Mill Top Cavity
Rough Main Hole Front Face	Drill 9 Holes Back Face	Rough Mill Piston Cylinders
Drill 4 Holes Front Face	Tap 9 Holes Back Face	Rough Mill Sides of Block
Drill 3 Holes Front Face	Drill 2 Holes Back Face	Rough Bottom of Block
Bore Large Hole Front Face	Counter Bore 2 Holes Back Face	Semi Rough Top Cavity
Drill 12 Holes Front Face	Drill 2 Holes Back Face	Semi Rough Piston Cylinders
Tap 12 Holes Front Face	Counter Bore 2 Holes Back Face	Semi Rough Sides of Block
Finish Front Face Square Corners	Drill 3 Holes Back Face	Semi Rough Bottom of Block
Finish Front Face Fillets	Counter Bore 3 Holes Back Face	Finish Top Cavity Square Corners
	Drill 5 Holes Back Face	Finish Top Cavity Fillets 1
	Finish Front Face Square Corners	Finish Top Cavity Fillets 2
	Finish Front Face Fillets	Finish Sides Square Corners
	Bore Main Hole Back Face	Finish Sides Fillets
		Finish Cylinder Counter Bore
		Finish Bottom Square Corners
		Finish Bottom Fillets
		Bore Cylinders to Finish
		Drill 35 Holes Around Cylinders
		Tap 35 Holes Around Cylinders
		Drill 12 Holes in Top Cavity
		Drill 1 Hole in Top Cavity
		Counter Bore 1 Hole in Top Cavity
		Drill 8 Holes Bottom of Block
		Tap 8 Holes Bottom of Block

The list of operations is then used to make specific assignments to each user so that all of the required operations are defined in the CAM software. The method of making assignments is described in the next section.

Operation Type Decomposition

Decomposition can also be performed based on the operation type. Table 4.2 helps to illustrate this method. It is also based on the engine block shown in Figure 4.2. All of the facing operations are identified, then the roughing operations are identified, then semi-roughing, then finishing, etc.

Similar to the previous method, this list is then used to assign operations to users. The users then create the operations in the CAM software.

It is important to note that the operation decomposition can also be completed in parallel. The users do not need to step through all the of the steps together. The part manager can assign different users a portion of the part to decompose into a list of operations independently of the other users. For example, using the machining order decomposition method, each user can be assigned a setup and decompose the operations needed to machine all the features for that setup. Similarly, using the operation type decomposition users can be split by operation type and identify all the operations of a specific type needed to machine the part. The list should then be compiled and a verification performed by the part manager or the whole group.

The beauty of these decomposition approaches is that no type of automatic subsystem is needed. However, a strategy is needed to assign proper expertise to each CAM task. Other research is being conducted that develops sophisticated knowledge databases to extract clients according to expertise and experience. This is necessary because multi-user CAx software will flatten the normal disciplinary hierarchies that still prevail in industry.

4.3.4 Assigning Tasks

At the completion of the part decomposition, each process step must be assigned to a user to be defined within the CAM software. A part manager in most cases assigns process steps to users, but this could also be done by the team. Assignments are made according to skills; users

Table 4.2: Operations for engine block shown in Figure 4.2

Operations		
Facing Operations	**Semi Rough Milling Operations**	**Drill and Tap Operations**
Face Front Face of Block	Semi Rough Front Face	Drill 4 Holes Front Face
Face Back Face of Block	Semi Rough Back Face	Drill 3 Holes Front Face
Face Mill Slanted Faces	Semi Rough Top Cavity	Drill 12 Holes Front Face
Face Mill Top Face	Semi Rough Piston Cylinders	Tap 12 Holes Front Face
Rough Milling Operations	Semi Rough Sides of Block	Drill 9 Holes Back Face
Rough Mill Front Face	Semi Rough Bottom of Block	Tap 9 Holes Back Face
Rough Main Hole Front Face	**Finish Milling Operations**	Drill 2 Holes Back Face
Rough Mill Back Face	Finish Front Face Square Corners	Drill 2 Holes Back Face
Rough Mill Top Cavity	Finish Front Face Fillets	Drill 3 Holes Back Face
Rough Mill Piston Cylinders	Finish Front Face Square Corners	Drill 5 Holes Back Face
Rough Mill Sides of Block	Finish Front Face Fillets	Drill 35 Holes Around Cylinders
Rough Bottom of Block	Finish Top Cavity Square Corners	Tap 35 Holes Around Cylinders
Bore Operations	Finish Top Cavity Fillets 1	Drill 12 Holes in Top Cavity
Counter Bore Large Hole Front Face	Finish Top Cavity Fillets 2	Drill 1 Hole in Top Cavity
Counter Bore 2 Holes Back Face	Finish Sides Square Corners	Drill 8 Holes Bottom of Block
Counter Bore 2 Holes Back Face	Finish Sides Fillets	Tap 8 Holes Bottom of Block
Counter Bore 2 Holes Back Face	Finish Cylinder Counter Bore	
Counter Bore 3 Holes Back Face	Finish Bottom Square Corners	
Counter Bore Main Hole Back Face	Finish Bottom Fillets	
Counter Bore Cylinders to Finish		
Counter Bore 1 Hole in Top Cavity		

who are experts in certain processes are assigned the operations relative to that expertise (MUCAM principle 6). This lowers the chance of needing to redefine an operation and ensures a high quality finish. All tasks can be assigned for synchronous performance by the assigned set of clients, or they can be assigned asynchronously depending on available workforce. Central to the success of either method is that the users communicate with one another to complete the task in the most effective manner. Of course, a client may be versed in several operations. This provides great flexibility in assigning operations to appropriate experts, expecting one or more experts to assemble the correct order of operations from all the client tool paths.

Figure 4.3 illustrates how the toolpath for a die is created by assigning a list of previously decomposed operations to users. The roughing operations are assigned to User 1, User 2 the finishing passes using, User n is assigned the finishing features (slotting, chamfering, etc) and so on. The users then work in parallel to create the operations within the CAM software as described in the next section.

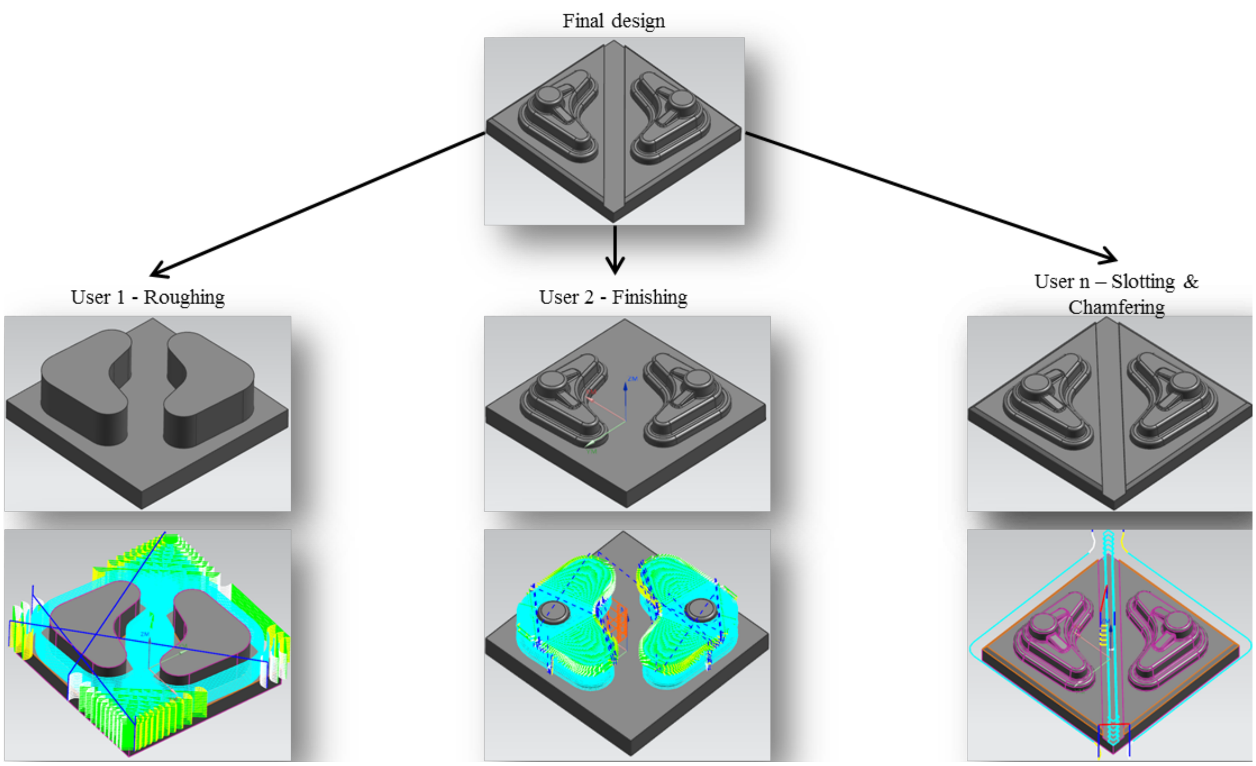


Figure 4.3: Decomposition of part into machining operations

4.3.5 Creation and Definition of Operations

At this point the users create and define the operations in the CAM software needed to complete the tasks assigned to them. Before the full toolpath is generated and exported as machine readable code it exists as a set these operations in the CAM software. Each operation is an object which contains a set of machining parameters as well as well as a list of faces/cavities to machine. These operations can be created in any order by the user. For example, a final cut can be defined by the user before the roughing cuts. The roughing cut is just simply inserted in the machining order before the final cut. When a user initiates the generation of the toolpath, the software steps through the operations in the order they were arranged and generates a full toolpath while at the same time tracking which material has already been removed from the stock piece. Later cuts ignore material that was already cut.

4.3.6 Final Compilation of Tasks

Once all operations and tools have been defined, all the users, except for the part manager, exit the session. The part manager then verifies and finalizes the process. As stated above, CAM software allows operations to be defined in any order within the software. The manager must verify that the order of operations is correct. If not, the manager reorders the operations into the proper sequence to ensure that the part will be machined correctly (MUCAM principle 5). The manager also performs a final simulation and any necessary checks to verify the machining processes. Once the manager is satisfied with the compiled program, it is post-processed and exported, ready to be loaded into one or more CNC machines.

For parts that require multiple machines, each program is exported individually by its manager and loaded onto the appropriate machine.

CHAPTER 5. EXPECTED BENEFITS

5.1 Introduction

As the research of this project has progressed, many significant potential benefits of a MUCAM system became apparent. The purpose of this section is to explain these benefits in depth to help the reader understand what features of multi-user CAM were being tested and observed in the testing phase of the project which is described in the next chapter.

5.2 Time Savings

The goal of all multi-user applications is to shorten the time to market by taking a process that is generally completed in a serial manner and creating a method for users to work in parallel. MUCAM has limitations that don't allow the entire process to be completed in parallel, but the main body of the work, done after setup and before post processing, can be done in parallel and thus achieve the goal of shortening the time to market.

An estimate of the time saved ΔT can be made in the following manner. It should be noted that the elapsed time and total man hours of work performed are equal for a single user. First, analyze how much time it takes for a single user to plan a process. The total time required for a single user to plan all the tool paths T_{SU} is simply the sum of the time required to plan the toolpaths t_{CAM}, the decomposition time t_{decomp} and the setup and finishing times, t_{setup} and t_{finish} respectively.

$$T_{SU} = t_{CAM} + t_{decomp} + t_{setup} + t_{finish} \tag{5.1}$$

The total elapsed time required for for a multi-user team to plan a process can be expressed as follows. The total work performed that can be divided among users is the sum of the time spent planning the toolpaths, represented by the term t_{CAM}, and the decomposition time t_{decomp}.

These terms are in fact equal to the terms t_{CAM} and t_{decomp} in the single-user case as the work that must be completed for a given part is equal in each case. However, in the multi-user case, these terms are divided by the number of users N and an efficiency factor e. The efficiency factor is included because there is a discreet set of tasks to perform that cannot be divided equally among the users as well as the small administrative time required to organize and divide the work among users. These terms are added to the setup and finishing times, t_{setup} and t_{finish} respectively. Note that this is a conservative estimate because, as discussed more later, in an optimal scenario expert users who are well versed in specific operations will complete the tasks more quickly than a single user who may be inexperienced in a subset of the required tasks. Thus, in an optimal scenario $(t_{cam} + t_{decomp})_{MU} < (t_{cam} + t_{decomp})_{SU}$.

$$T_{MU,elapsed} = \left(\frac{t_{CAM} + t_{decomp}}{Ne} \right) + t_{setup} + t_{finish} \tag{5.2}$$

The man hours required for a multi-user team to plan a process can be expressed similarly:

$$T_{MU,manhours} = \frac{(t_{CAM} + t_{decomp})}{e} + t_{setup} + t_{finish} \tag{5.3}$$

In an ideal case (an efficiency of 1.0 and work that is perfectly divisible between the users) equations 5.1 and 5.3 would be equal.

The efficiency factor e can be determined by comparing the man hours of the single user with the man hours spent by the multi-user team:

$$e = \frac{(t_{CAM} + t_{decomp})_{SU}}{(t_{CAM} + t_{decomp})_{MU}} \tag{5.4}$$

The elapsed time saved ΔT is the difference between the single user time described in equation 5.1 and the multi-user time described by equation 5.2.

$$\Delta T_{elapsed} = (t_{CAM} + t_{decomp}) \left(1 - \frac{1}{Ne} \right) \tag{5.5}$$

The difference in man hours ΔT is the difference between the single user time described in equation 5.1 and the multi-user time described by equation 5.3.

$$\Delta T_{manhours} = (t_{CAM} + t_{decomp}) \left(1 - \frac{1}{e}\right) \tag{5.6}$$

Equation 5.5 shows that the elapsed time saved, ΔT, is a function of the number of users. By increasing the number of users, the time saved increases, although generally it can be expected that the efficiency will decrease. This reflects a statement made by the NXConnect developers: "This new distributed workflow enables fully realized product designs at a much faster pace than was previously possible." [**?**]. A test was used to demonstrate the time savings and is described fully in chapter 6.

5.3 Expert Users

One of the features of MUCAM is the ability to involve expert users (MUCAM principle 6). Often users are not experts in all manufacturing processes. However, because multiple users are working on the part, each operation can be assigned to the user most knowledgeable in that specific process. The main benefit of this is that it reduces the likelihood of errors in the creation of the operation thus avoiding any rework of the operation.

The use of experts leads to other benefits. Experts will plan operations to run in the most efficient way. Also, their knowledge allows them to define the operation in the software in a more quick and decisive manner as well.

5.4 Training

When clients engage the model simultaneously there is a unique opportunity for experts to train novices in new capabilities (MUCAM principle 7). This is currently not possible with existing CAM applications, other than looking over the shoulder of another user. Mixing expert users with non-expert users creates an ideal training environment. MUCAM allows effective knowledge transfer from expert users to novices.

5.5 Oversight

MUCAM has the potential to eliminate mistakes by multi-user observation and oversight. A single user working on a complex CAM setup can make mistakes that may go unnoticed for a long period of time. In a multi-user environment there are multiple users constantly looking over the program and communicating with each other. There is also a part manager that provides oversight. With the involvement of so many users, mistakes are more likely to be caught early in the process.

5.6 Summary

Each of these benefits serves to decrease the time to market of a product that is under development. There is also the potential to increase the quality of the final product. MUCAM software allows these benefits to be realized. However, MUCAM is not feasible without implementing the decomposition method to organize the work from the outset of the tool path planning stage of the project.

CHAPTER 6. TESTING

A MUCAM prototype was developed based on the methods described in chapter 3. The following sections describe how the decomposition method that is described in chapter 4 was tested using the MUCAM prototype.

6.1 Method

The main goals of the tests described below are to show that multi-user CAM is a feasible tool and then to validate the decomposition method developed for use with MUCAM. Initially the test was completed using single-user CAM, after which it was completed using MUCAM. In each test tool paths were developed for the part shown in Figure 6.1. This part is a simplified version of an engine block.

Figure 6.1: Engine block used for testing MUCAM

It is understood by the author that this is a simple case and does not reflect the best applications for MUCAM. Many large scale scenarios exist where the CAM portion of the development process requires hundreds of hours of CAM work. This testing was done in an academic setting and thus was restricted to a smaller test case. Expert users were not available for the MUCAM testing, so intermediately skilled users worked together on the MUCAM team.

6.1.1 Single-user Testing

The part model was taken to the *Precision Machining Lab* at BYU. Using Gibbs CAM, a well known CAM application, an experienced user developed the toolpaths for the engine block. The user followed the method described in section 2.1.1. Specifically, after the initial setup steps were completed, the part was analyzed to see which operation should be completed first. The user then created (or selected those which had already been created) the supporting objects: Program, Tool, Geometry, Method. That operation was created and verified. The user then looped back to choose the next operation based on the remaining material. This process looped until all of the excess material had been addressed.

This method was used to generate tool paths for the part for each of the 5 different setups required to machine the part. The user identified the tools needed to manufacture the part while creating the operations. The list of tools is shown in Table 6.1.

Table 6.1: Tools used to machine the engine block shown in Figure 6.1

Tools
.735" Drill
3/4" Reamer
.075" Drill
1/8" End Mill
1/4" End Mill
1/8" Ball End Mill
1/8" Drill
3/16" Ball End Mill
3/4" End Mill
1/2" End Mill

The operations generated by the single user to machine the part are shown in Table 6.2. The PML reported spending approximately 7 hours to complete the decomposition and generation of the tool paths for the part. It is also of interest that the assigned person took several months to complete this project. It kept getting bumped by higher priority projects. In industrial settings this same thing can occur.

Table 6.2: Operations generated by single-user to machine engine block shown in Figure 6.1

Operations		
Transmission End	**Front End**	**Right Cylinders**
.735" Drill Camshaft	1/8" Drill Holes	3/4" End Mill Rough Cut Cylinders
.735" Drill Crankshaft	1/4" End Mill Rough Cut	3/4" End Mill Finish Cut Cylinders
3/4" Ream Camshaft	3/16" Ball Mill Finish	1/2" End Mill Rough Top Cavity
3/4" Ream Crankshaft	**Bottom End**	1/8" Ball Mill Semi-finish Top Cavity
.075 Drill Holes	1/2" End Mill Face Bottom Surface	1/8" Ball Mill Finish Top Cavity
1/8" End Mill Outside Oval	1/2" End Mill Cavity	**Left Cylinders**
1/4" End Mill Inside Oval	1/2" End Mill Profile Cut Ledges	3/4" End Mill Rough Cut Cylinders
1/8" Ball Mill Profile Cut		3/4" End Mill Finish Cut Cylinders
		1/2" End Mill Rough Top Cavity
		1/8" Ball Mill Semi-finish Top Cavity
		1/8" Ball Mill Finish Top Cavity

6.1.2 MUCAM Testing

Testing using the MUCAM prototype proceeded as described in chapter 4. It is important to fully describe the circumstances of the test. Expert users were not available so 4 users with CAM experience were selected from the available group. The users were considered intermediate users of CAM software. As this is the first known attempt at MUCAM, the users can all be considered novices in MUCAM. Although, this setup is not optimum, it still allows a suitable testing environment for the main question to be answered by this research: Is it possible for several users to simultaneously generate tool paths for a single part.

Initially three of the four students met together to begin the MUCAM process. A single user was assigned to be the part manager (MUCAM Principle 1). Next the team together began the decomposition step. Each user had the part open on their own computer and the discussion began. The first 30 minutes were spent exploring the part and generating a tool list based on the size of features as well as getting an idea of what operations would be needed. The tool list they generated is shown in Table 6.3.

Table 6.3: Tools used to machine the engine block shown in Figure 6.1

Tools
3/4” Drill
.075” Drill
1/8” Drill
1/8” End Mill
3/8” Ball End Mill
45° Chamfer Tool
1/8” Ball End Mill
1/2” Bull Nose Mill
1/4” End Mill

As they worked, they identified 5 different setups that would be required to manufacture the part. Together they decided to approach decomposition using the Machining Order Decomposition method described in section 4.3.3 Dividing the work according to the five setups seemed like an obvious choice so each user was assigned a setup for which they were to identify the operations to be created for that setup in the order they would be machined. The users then worked to generate the complete list of operations setup by setup. As they worked, they discussed various problems and concerns, using the other group members' knowledge when needed. The list of operations that was generated during the decomposition discussion is shown in Table 6.4.

At the end of the decomposition step a relatively short discussion was held to verify the planned operations and to assign users to generate the operations within the CAM software. Two users were assigned to generate the tool paths for the right and left side cylinders and the top cavity. Users 3 and 4 worked on the front and transmission ends respectively (MUCAM Principle 3). The bottom end was left to be completed by user 1 at the completion of the other operations. The team was assigned tasks based on skill where possible. Most significantly, user 1 had experience with 3D surfacing and performed much of the work on the areas with 3D faces (MUCAM Principle 6).

The team was then ready to generate the operations in the MUCAM software. The part manager opened the part in the MUCAM application, selected the 3-axis mill environment, identified the setups needed and then setup all of the reference geometry: the coordinate systems and stock pieces for each of the 5 setups. The users then worked to generate the operations based on their assignments as listed in Table 6.4.

This portion of the testing was the most significant test of the actual software. As can be expected errors interrupted the workflow. However, only one type of error was encountered: a loss of a message from the server to a client. This would cause that user to be out of sync with the other users. This required all the users to pause and reload from a previously saved state and then redo some of the work they had already completed. These errors were accounted for in the timing of the test.

At this point all of the operations had been created. All of the users exited except the part manager. The part manager performed a final verification of all of the tool paths and checked the operation order (MUCAM Principle 5). Also the manager was able to generate the proper stock

Table 6.4: Operations and user assignments generated by the multi-user team to machine engine block shown in Figure 6.1

Operations	**User**
Transmission End	
3/4” Drill Camshaft	4
3/4” Drill Crankshaft	4
.075” Drill Holes	4
1/4” End Mill Rough Cut	4
1/8” End Mill Finishing Cut	4
1/8” Ball Mill Profile Cut	4
Front End	
1/8” Drill Holes	3
1/4” End Mill Rough Cut	3
1/8” End Mill Finishing Cut	3
1/8” Ball Mill Profile Cut	3
Right Cylinders	
1/2” Bull Nose Mill Rough Cut Cylinders	2
1/2” Bull Nose Mill Finish Cut Cylinders	2
3/8” Ball Mill 3D Surfacing Top Cavity	1
1/8” Ball Mill 3D surfacing Top Cavity	1
45° Chamfer Mill Finish Surfacing	1
Left Cylinders	
1/2” Bull Nose Mill Rough Cut Cylinders	2
1/2” Bull Nose Mill Finish Cut Cylinders	2
3/8” Ball Mill 3D Surfacing Top Cavity	1
1/8” Ball Mill 3D surfacing Top Cavity	1
45° Chamfer Mill Finish Surfacing	1
Bottom End	
1/2” Bull Nose Mill Rough Cut Cavity	1
3/8” Ball Mill Finish Bottom 3D Surface	1

Table 6.5: Time contributions by users during MUCAM testing.

Session	User 1	User 2	User 3	User 4	Total
Decomposition	1:00	1:00	1:00	0:00	**3:00**
Setup	0:30	0:00	0:00	0:00	**0:30**
Operation Generation	0:30	0:00	0:30	0:30	**1:30**
Operation Generation	1:00	1:00	0:00	0:30	**2:30**
Operation Generation	1:00	1:00	0:00	0:00	**2:00**
Verification/Code Generation	0:30	0:00	0:00	0:00	**0:30**
Total Hours of Labor	4:30	3:00	1:30	1:00	10:00

pieces for each setup that correctly represented the material left after the previous steps. The final tool paths were then generated, exported, and loaded into a CNC 3-axis mill.

The CNC 3-axis mill was used to machine the part from an aluminum block. The final result is shown in figure 6.2. The entire tool path planing process for this part was completed in an elapsed time of 4.5 hours by the MUCAM team. This time is an approximation based on the time users were working with adjustments for software errors and administrative activities. The experiment was completed in 6 sessions with various users present at different times. Combining the participation time of each user results in a total of 10 hours of labor. The participation is laid out in Table 6.5.

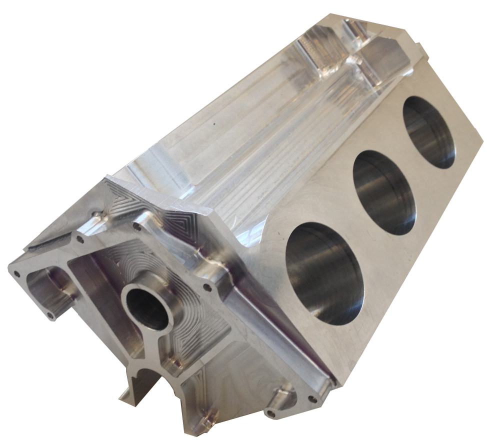


Figure 6.2: Engine block machined using tool paths generated by MUCAM team

CHAPTER 7. ANALYSIS

7.1 Multi-User CAM

This research represents the first known attempt at multi-user tool path planning. As stated in the problem statement, the first question to be answered is the following: Is it possible for several users to simultaneously generate tool paths for a single part in a parallel workflow? It was shown in the previous chapter that it is indeed possible for several users to work in parallel to generate tool paths. It was shown in figure 4.1 that there is work at the beginning and end that must be completed by a single user. However, the time consuming portion of tool path planning can indeed be worked on in parallel by several users simultaneously.

7.2 Time Savings

It was theorized in chapter 5 that the elapsed time saved, ΔT, from using a multi-user team for tool path planning can be calculated using equation 5.5 which is shown again here:

$$\Delta T_{elapsed} = \left(t_{CAM} + t_{decomp}\right)\left(1 - \frac{1}{Ne}\right) \tag{5.5 revisited}$$

To apply this equation the efficiency factor must first be calculated. The single user reported 7.0 hours spent to complete t_{CAM} and t_{decomp}. The data shown in Table 6.5 shows the multi-user team spent 9.0 man hours to complete t_{CAM} and t_{decomp} and an elapsed time of 3.5 hours. The efficiency factor can be calculated using 5.4 which is shown here again:

$$e = \frac{\left(t_{CAM} + t_{decomp}\right)_{SU}}{\left(\left(t_{CAM} + t_{decomp}\right)_{MU}} \tag{5.4 revisited}$$

Applying the man hours spent in each case to equation 5.4 results in an efficiency factor of 0.78. It is important to note that although four users participated in the multi-user case, four users were never working simultaneously. Table 6.5 shows which users were present during which sessions and for how long. By breaking the sessions into 0.5 hour sessions and averaging the number of users present in each session results in an average of 2.57 users. Applying these numbers to equation 5.5 results in 3.5 hours saved which also results in an estimate of the multi-user team spending 3.5 hours to complete the task, which matches the test data.

As noted before, the multi-user team completed the project over a few days. If not for software errors and administrative activities the project would have been completed even more quickly. The single-user worked on the project in short bursts over several months as projects that were considered higher priority were assigned. Although this phenomenon was not directly tested, this behavior provides additional insight into methods for shortening time to market. It appears that a project that requires team of users creates a certain accountability the helps a project to be completed in a shorter amount of time (start to finish). Stopping and restarting a project often requires extra time for the user to reacquaint themselves with the project. Also, it is easy for a project to get lost among a never ending flow of higher priority projects.

In the testing that took place we found that the single-user generated relatively the same list of operations that were generated by the MUCAM team, as shown in table 7.1. This shows that the methods used to decompose the parts in many cases are relatively obvious, even to less experienced users. Thus, they can be relied on as effective methods by multi-user teams.

7.3 Expert Users

The users on the multi-user team cannot be considered expert users. However, there were clear differences in knowledge from one user to the next during testing. Thus, idea of assigning an expert to specific tasks was still relevant (MUCAM Principle 6). For example, users who had experience in the more difficult task, 3D surfacing, were assigned to complete the 3D surfacing operations. The users experienced in 3D surfacing knew exactly how to set the parameters to generate a nicely contoured surface. The other users would have been able to do it but it would have required time to research and learn the proper methods.

Table 7.1: Comparison of operations generated by a single user and a multi-user team

Multi-user Team	Single-user
Transmission End	**Transmission End**
3/4" Drill Camshaft	.735" Drill Camshaft
3/4" Drill Crankshaft	.735" Drill Crankshaft
.075" Drill Holes	3/4" Ream Camshaft
1/4" End Mill Rough Cut	3/4" Ream Crankshaft
1/8" End Mill Finishing Cut	.075 Drill Holes
1/8" Ball Mill Profile Cut	1/8" End Mill Outside Oval
	1/4" End Mill Inside Oval
	1/8" Ball Mill Profile Cut
Front End	**Front End**
1/8" Drill Holes	1/8" Drill Holes
1/4" End Mill Rough Cut	1/4" End Mill Rough Cut
1/8" End Mill Finishing Cut	3/16" Ball Mill Finish
1/8" Ball Mill Profile Cut	
Right Cylinders	**Right Cylinders**
1/2" Bull Nose Mill Rough Cut Cylinders	3/4" End Mill Rough Cut Cylinders
1/2" Bull Nose Mill Finish Cut Cylinders	3/4" End Mill Finish Cut Cylinders
3/8" Ball Mill Rough Top Cavity	1/2" End Mill Rough Top Cavity
1/8" Ball Mill Contour Top Cavity	1/8" Ball Mill Semi-finish Top Cavity
45° Chamfer Mill Finish Surfacing	1/8" Ball Mill Finish Top Cavity
Left Cylinders	**Left Cylinders**
1/2" Bull Nose Mill Rough Cut Cylinders	3/4" End Mill Rough Cut Cylinders
1/2" Bull Nose Mill Finish Cut Cylinders	3/4" End Mill Finish Cut Cylinders
3/8" Ball Mill Rough Top Cavity	1/2" End Mill Rough Top Cavity
1/8" Ball Mill Contour Top Cavity	1/8" Ball Mill Semi-finish Top Cavity
45° Chamfer Mill Finish Surfacing	1/8" Ball Mill Finish Top Cavity
Bottom End	**Bottom End**
1/2" Bull Nose Mill Rough Cut Cavity	1/2" End Mill Face Bottom Surface
3/8" Ball Mill Finish Contour Cavity	1/2" End Mill Cavity
	1/2" End Mill Profile Cut Ledges

7.4 Training and Oversight

The lack of experts on the team during testing lead to a lot of training and oversight activity (MUCAM principle 7). Different users had knowledge in different areas and the knowledge was shared with other users as needed. The users were working together in the same room on separate computers. When a user had a question, the group together would discuss the best methods and how to proceed. This impromptu training resulted in users learning more about tool paths for difficult features and best practices for generating tool paths. Apart from decreasing time to market, this is probably the most significant feature of MUCAM. Expert users are able to work with less experienced users to pass along knowledge and skill in context while completing a problem.

CHAPTER 8. CONCLUSIONS AND RECOMMENDATIONS

8.1 Summary

Multi-user CAM was developed to reduce the time to market by involving multiple users in the definition of tool path processes. Current CAM applications force a strictly serial approach to tool path planning. These serial processes are contrary to collaborative principles that development teams apply to bring a product to market. It was shown that multi-user CAM allows users to work in parallel and thus provides an environment where collaborative principles used by development teams can be extended to the CAM software they use.

Strong clients and a thin server are the optimal solution for the implementation of a multi-user CAM system. As users perform actions within MUCAM, the data is passed to the server where it is stored and then passed on to all other connected clients. This allows a real-time collaboration environment. The MUCAM application was developed in such a way that the user will see basically no difference between the single user software they are used to and the MUCAM application interface.

The MUCAM Principles laid out and in section 4.2 and verified through testing are central to the success of multi-user tool path planning. Of specific import is the natural way that a process decomposes into a set of operations that can be defined individually. An operation is comprised of all the movements that can be completed with a set of defined parameters. Any change in a parameter requires the creation of a new operation. The set of operations are identified by the team before beginning and are then assigned to individual users.

8.2 Conclusions

Although the sample size is 1, the data and analysis seems indicate that the time saved is a function of the number of users. These estimates of time savings were found to be comparable to multi-user design software which has been fully developed and implemented. Therefore, the goal of shortening the time to market can be achieved by implementing a multi-user CAM environment. To remain competitive, industries must be able to shorten the time to market to gain an economic edge. It seems that the obvious conclusion is to recommend the full development and implementation of a multi-user CAM environment.

This research also successfully completed all of the research objectives enumerated in Chapter 1:

- Develop and program Multi-user CAM functionality into NXConnect.
- Develop a method to distribute operationally independent feature groups among team members.
- Use Multi-user CAM to verify the method of distribution among users.
- Use software to plan the tool paths for a part and manufacture it as a final proof of concept.

The successful completion of these objectives shows that it is indeed possible for several users to simultaneously generate tool paths for a single part in a parallel workflow. Principles and processes were developed and described in Chapter 4 that can be used to decompose a manufacturing process into a series of steps that can be entered into CAM software simultaneously by multiple users.

8.3 Recommendations for Future Work

MUCAM is in its earliest stages and thus a lot of work remains. The MUCAM prototype only supported the following operation types within the NX manufacturing module.

- Face Milling Area
- Face Milling Manual
- Cavity Mill
- Contour Area

- Contour Profile
- Corner Rough
- Z Level Profile
- Flow-cut Ref Tool

Support for the rest of the operations in NX needs to be coded into the software.

All of the tools available in NX Manufacturing are supported, but currently Programs, Methods, and supporting geometry are not supported. A single user can create these objects, but they currently are not passed across the server to other users. To develop tool paths in a multi-user setting these do not have to be supported, but to create robust software this feature must be added.

There is of course lots of room to improve the user experience withing MUCAM. For example, standard features such as Undo and the proper deletion of objects need to be added.

Aside from the actual software, research is needed to better identify which parts are candidates for the Multi-User CAM approach to tool path planning. Potential indicators were enumerated in 4.2.2 but it would require further research to verify this estimation.

REFERENCES

[1] Red, E., Jensen, G., French, D., and Weerakoon, P., 2011. "Multi-user architectures for computer-aided engineering collaboration". In Concurrent Enterprising (ICE), 2011 17th International Conference on, IEEE, pp. 1–10. vii, 7, 15

[2] Sahov, J., 2015. Engine block, cad model, May. vii, 24

[3] Red, E., Holyoak, V., Jensen, C. G., Marshall, F., Ryskamp, J., and Xu, Y., 2010. "ν-cax: A research agenda for collaborative computer-aided applications". *Computer-Aided Design and Applications,* **7**(3), pp. 387–404. 1, 10

[4] Xu, Y., 2011. "A flexible context architecture for a multi-user gui.". Master's thesis, Brigham Young University. 1, 11, 14

[5] Hepworth, A. I., Tew, K., Trent, M., Ricks, D., Jensen, C. G., and Red, E., 2014. "Model consistency and conflict resolution with data preservation in multi-user computer aided design". *Journal of Computing and Information Science in Engineering,* **14**(2). 1

[6] Red, E., Jensen, G., Weerakoon, P., French, D., Benzley, S., and Merkley, K., 2013. "Architectural limitations in multi-user computer-aided engineering applications.". *Computer & Information Science,* **6**(4). 1

[7] Marshall, F., 2011. "Model decomposition and constraints to parametrically partition design space in a collaborative cax environment.". Master's thesis, Brigham Young University. 2, 10, 11, 12

[8] Corp., U., 2007. *NX Manufacturing Fundementals*. 3, 4

[9] Okulicz, K., 2004. "Virtual reality-based approach to manufacturing process planning". *International journal of production research,* **42**(17), pp. 3493–3504. 7

[10] Dong, J. J., Parsaei, H. R., and Leep, H. R., 1996. "Manufacturing process planning in a concurrent design and manufacturing environment". *Computers & industrial engineering,* **30**(1), pp. 83–93. 7

[11] Hong-Tzong, Y., and Chia-Hsiang, M., 1993. "Concurrent process planning for finish milling and dimensional inspection of sculptured surfaces in die and mould manufacturing.". *International Journal of Production Research,* **31**(11), p. 2709. 7

[12] Sun, C., Xia, S., Sun, D., Chen, D., Shen, H., and Cai, W., 2006. "Transparent adaptation of single-user applications for multi-user real-time collaboration". *ACM Transactions on Computer-Human Interaction (TOCHI),* **13**(4), pp. 531–582. 8

[13] Google, 2015. Drive help: Limits on sharing. 9

[14] French, D. J., Stone, B., Nysetvold, T. T., Hepworth, A., and Red, W. E., 2014. "Collaborative design principles from minecraft with applications to multi-user cad". In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V01BT02A039–V01BT02A039. 10

[15] Ming, X. G., Yan, J. Q., Wang, X., Li, S., Lu, W. F., Peng, Q., and Ma, Y., 2008. "Collaborative process planning and manufacturing in product lifecycle management". *Computers in Industry,* **59**(2), pp. 154–166. 10

[16] Sarma, S. E., and Wright, P. K., 1996. "Algorithms for the minimization of setups and tool changes in simply fixturable components in milling". *Journal of Manufacturing Systems,* **15**(2), pp. 95–112. 11

[17] Shah, J. J., Mäntylä, M., and Nau, D. S., 2013. *Advances in feature based manufacturing*. Elsevier. 11

[18] Woo, T. C., 1982. "Feature extraction by volume decomposition". In Conference on CAD/CAM technology in mechanical engineering, pp. 76–94. 11

[19] Kim, Y. S., 1991. "Form feature recognition by convex decomposition". *Procs ASME, DETC/CIE*, pp. 61–71. 11

[20] Kim, Y. S., and Wilde, D., 1992. "A convergent convex decomposition of polyhedral objects". *Journal of Mechanical Design,* **114**(3), pp. 468–476. 11

[21] Perng, D.-B., Chen, Z., and Li, R.-K., 1990. "Automatic 3d machining feature extraction from 3d csg solid input". *Computer-Aided Design,* **22**(5), pp. 285–295. 11

[22] Jared, G. E., 1984. "Shape features in geometric modeling". In *Solid Modeling by Computers*. Springer, pp. 121–137. 11

[23] Kyprianou, L. K., 1980. "Shape classification in computer-aided design". PhD thesis, University of Cambridge. 11

[24] Jakubowski, R., 1986. "A structural representation of shape and its features". *Information sciences,* **39**(2), pp. 129–151. 11

[25] Choi, B., 1985. "Stopp: an approach to cadcam integration". *Computer-Aided Design,* **17**(4), pp. 162–168. 11

[26] Srinivasan, R., Liu, C., and Fu, K., 1985. "Extraction of manufacturing details from geometric models". *Computers & industrial engineering,* **9**(2), pp. 125–133. 11

[27] Joshi, S., and Chang, T.-C., 1990. "Feature extraction and feature based design approaches in the development of design interface for process planning". *Journal of Intelligent Manufacturing,* **1**(1), pp. 1–15. 11

[28] Woodwark, J., 1988. "Some speculations on feature recognition". *Computer-Aided Design,* **20**(4), pp. 189–196. 11

[29] Lee, Y., and Jea, K., 1988. "A new csg tree reconstruction algorithm for feature representation". In Computers in Engineering Conference (CIE), pp. 521–528. 11

[30] Shpitalni, M., and Fischer, A., 1991. "Csg representation as a basis for extraction of machining features". *CIRP Annals-Manufacturing Technology,* **40**(1), pp. 157–160. 11

[31] Lee, Y.-C., and Jea, K., 1990. "Feature extraction based design retrieval". In Systems Integration, 1990. Systems Integration'90., Proceedings of the First International Conference on, IEEE, pp. 214–223. 11

[32] Lee, Y., and Fu, K., 1987. "Machine understanding of csg: extraction and unification of manufacturing features". *Computer Graphics and Applications, IEEE,* **7**(1), pp. 20–32. 11

[33] Pande, S., and Prabhu, B., 1990. "An expert system for automatic extraction of machining features and tooling selection for automats". *Computer-Aided Engineering Journal,* **7**(4), pp. 99–103. 11

[34] Marefat, M., and Kashyap, R., 1990. "Geometric reasoning for recognition of three-dimensional object features". *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* **12**(10), pp. 949–965. 11

[35] Joshi, S. B., 1987. Cad interface for automated process planning. Tech. rep., Purdue Univ., Lafayette, IN (USA). 11

[36] Arikan, M. S., and Totuk, O. H., 1992. "Design by using machining operations". *CIRP Annals-Manufacturing Technology,* **41**(1), pp. 185–188. 11

[37] Wei, Y., and Egbelu, P. J., 2000. "Process alternative generation from product geometric design data". *IIE transactions,* **32**(1), pp. 71–82. 11

[38] Yahia, N., Fnaiech, F., Abid, S., and Sassi, B., 2002. "Manufacturing process planning application using artificial neural networks". Vol. 5, IEEE International Conference on Systems, Man and Cybernetics. 11

[39] Rho, H., and Lee, C., 1996. "Manufacturing features applied to the milling operation planning.". 3rd CIRP Workshop on Design and Implementation of Intelligent Manufacturing Systems (IMS). 12

[40] Kim, Y. S., Wang, E., and Rho, H. M., 2001. "Geometry-based machining precedence reasoning for feature-based process planning". *International Journal of Production Research,* **39**(10), pp. 2077–2103. 12