COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 5, 675–683
http://dx.doi.org/10.1080/16864360.2016.1150714

Multi-user collaborative tool path planning using process decomposition

Andrew S. Priddis and W. Edward Red

Brigham Young University, USA

ABSTRACT
Single-user computer-aided engineering applications force a strictly serial design process, which ultimately lengthens time to market. We created multi-user tool path planning software, which allows the work to be done in parallel by several users who are experts in various processes. Key to decreasing the time spent developing tool paths is the decomposition of the various operations needed to manufacture a part among various users. By observing the use of other multi-user applications as well as observing users developing tool paths using current CAM software, we developed a method to decompose the paths among the users. Finally, we estimate the process time saved.

KEYWORDS
Multi-user; collaboration; CAM; tool path planning

1. Introduction

Red [6] notes that single-user tools constrain developers in collaborative work environments at every step of the product development process, from idea generation to manufacture. Single-user CAx applications like CAD, CAE, and CAM force a strictly serial design process, which ultimately lengthens time to market. These serial processes are contrary to collaborative principles that development teams apply to bring a product to market. New multi-user applications such as NXConnect [10] enable users to work in parallel during the design stage of the product development. We developed multi-user collaborative tool path planning software to rectify the current serial limitations in tool path planning, thereby decreasing cost and increasing the quality of manufacturing processes.

Modern large scale CAD packages integrate manufacturing tools that assist in process planning; sometimes, third party applications are used. 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. 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.

Multi-user CAM process planning addresses part complexity by involving multiple users, sometimes referred to as clients, in the workflow as they work either synchronously (simultaneously), or asynchronously using decomposition principles to divide the work when it is mostly independent. Parallel workflow also means that relevant tasks can be assigned to users who are experts in various processes. Using this approach increases quality, decreases cost and reduces lead times. The developers of NXConnect [8] have shown that "collaboration decreases the product development time in proportion to the number of multi users." This research will demonstrate similar time savings that accrue to multi-user CAM process planning.

Other researchers have recognized the need for a multi-user environment for product development. Hepworth [1] lists ten different multi-user CAD systems that have been developed to address this need. By employing the same approach to manufacturing process planning a similar improvement in efficiency is expected, enabling companies to be more competitive.

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

CONTACT W. Edward Red ered@byu.edu

assignments among several users. Tool path planning is unique in that the processes naturally decompose into distinct activities that can be divided among users. The processes can be defined independently of each other with little regard for what other users have done. There has been some research conducted in decomposition of features in multi-user modeling CAD, but a literature review has not discovered similar research in multi-user tool path planning.

2. State of current software

2.1. Tool path planning: single user

2.1.1. Overview

Multi-user tool path planning is completed using processes similar to those used in typical single user tool path planning. The processes used to develop tool paths by a single user are described here to aid the reader in understanding how the process differs from the process used by multiple users to develop tool paths. Multi-user process variations will be described in the following section. Details of the single user process are taken from the NX CAM training manual. [4]

Figure 1 [4] 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 1 shows that the definition of parent groups can be completed in any order, but the work proceeds in a serial fashion. The process loops after an operation is created until every operation has been defined. It will be shown later that

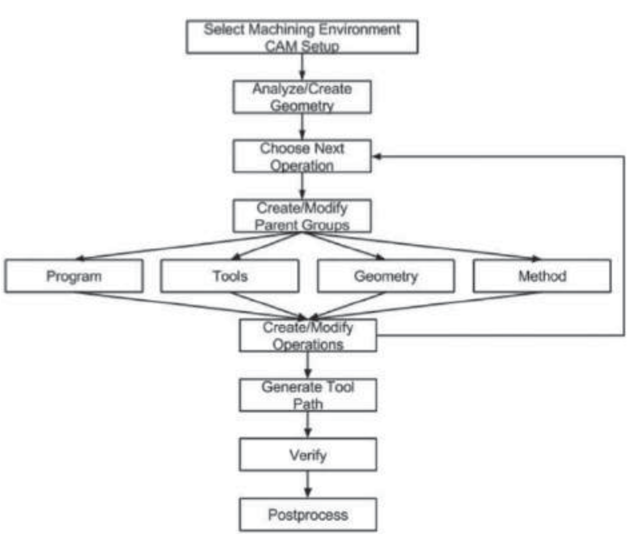

Figure 1. Single user tool path planning loops until every operation has been defined.

multi-user CAM differs by removing the loop. 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 step before proceeding in the process.

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, 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 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 very 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 in any order but must reordered to the correct sequence before 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 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 Fig. 2. 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.

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 for a set of operations that would be performed on the same machine at the same time. 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."[5] 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 [7] 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." In a collaborative design process this creates a significant bottleneck in the overall design process.

2.2. Multi-user applications

Research on different types of collaborative engineering applications has been conducted by several groups of researchers. Ming et al. [3] 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. Through collaboration 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. Our multi-user CAM differs by allowing multiple users to collaborate within the same CAM session. Implementing this multi-user CAM environment will result in increased productivity in the same areas.

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. [6] In a 2011 paper, Xu [10] describes the architecture and implementation of NXConnect, which is to be 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."

Much of the research conducted using NXConnect is especially applicable to decomposition for multi-user tool path planning. Marshall [2] used NXConnect to research how to decompose a part into a set of features and regions that can be modeled simultaneously by multi users. 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

Figure 2. Typical serial progression of operations for an engine block.

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.

3. Multi-user tool path planning prototype

3.1. Overview

The Multi-user CAM software we created differs greatly from single user CAM because several users can engage the model and plan tool paths at the same time. The first multi-user tool path planning prototypes are designed to run on the NXConnect architecture. [10] 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 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. 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 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. Therefore, a new process was created. The process is described in the following sections.

3.2. 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 (MUCAM) requires planning on several levels to be successful. First the users must plan which machine(s) to use. Later they will choose which operations to use on each machine.

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.

3.3. 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.

To most effectively setup the part and also complete the following process steps it is important to assign a part manager. The 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.

3.4. 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.

NC machining and CAM software are unique in that there is a very natural decomposition method. 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 operations 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 these steps that can each be defined independently of the others.

The beauty of this approach 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.

Figure 3 illustrates how a die can be machined by decomposing the operations into roles for several users. In this example all of the roughing passes are completed in a single operation, so user 1 defines the roughing operation. User 2 completes the finishing passes using two or three operations, which differ by tool used. The final user is assigned the finishing features and uses several operations to define them.

3.5. Assigning tasks

At the completion of the part decomposition, the users must determine who will define which operations and the related process steps and machine assignment. Users who are experts in certain processes are assigned the operations relative to that expertise. 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 of the client tool paths.

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 final process. As stated above, CAM software allows operations to be defined in any order within the software and then they must be appropriately reordered for final production. 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. 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.

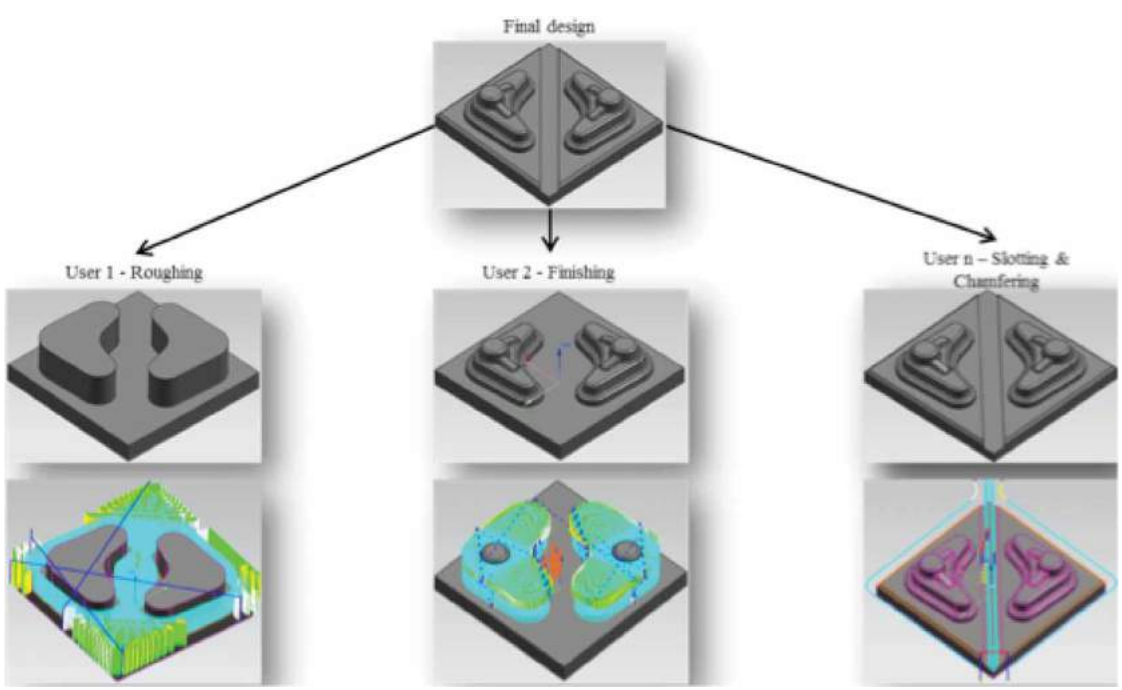

Figure 3. Decomposition of part into machining operations.

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

3.7. MUCAM prototype

3.7.1. Architecture

The MUCAM prototype is based upon the NXConnect Architecture described by Red et al. As shown in Fig. 4(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." [7]

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.

3.7.1.1. Client. When the MUCAM application is loaded, the modules shown in Fig. 4(b) are enabled. The Data Capture Module monitors the user's 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 user's object tree.

3.7.1.2. 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.

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 string of parameter names and values and is entered as a single line in the database table. These messages can hold hundreds of parameters in a single message. By storing

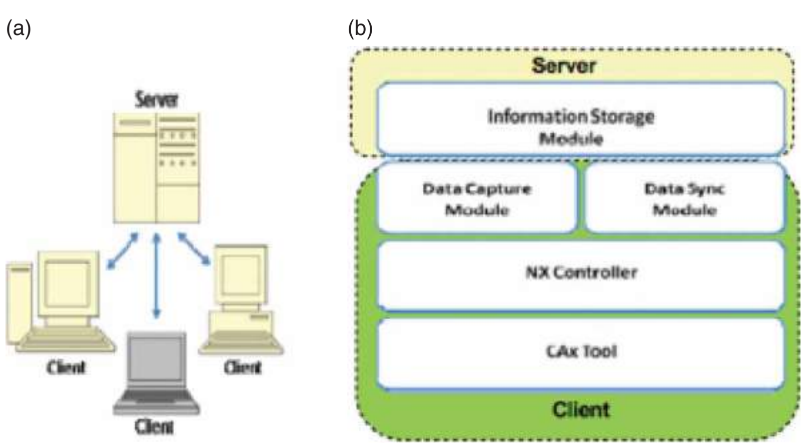

Figure 4. NXConnect architecture [7].

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, and once again allows the heavy work to be performed by the clients. These long strings are generated on the client by passing a CAM object into a serialization library before sending it to the server. Received messages are passed back into the serialization library and are returned as a CAM object.

3.7.2. 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 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.

4. Benefits

4.1. 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 a few 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.

The time savings can be quantified in the following manner. First we analyze how much time it takes for a single user to plan a process. Each operation is assigned a weighting factor, w. This weighting factor is multiplied by the average time it takes to complete an operation. The total time to plan a process is a function of the number of operations, the average time to plan a single operation, the weighting factor, and the setup and finishing times.

$$T_{Total} = \left(\sum w_i * t_{task,avg}\right) + t_{setup} + t_{finish} \qquad (4.1)$$

The time needed for a team to plan a process can be analyzed as follows. If it is assumed that each user has the same level of expertise then the total time is a function of the number of operations, the number of users, the average time to plan a single operation, the weighting factor, and the setup and finishing times.

$$T_{Total} = \frac{\left(\sum w_i * t_{task,avg}\right)}{N_{users}} + t_{setup} + t_{finish} \qquad (4.2)$$

The time savings is the difference between the single user time and the multi-user time.

$$\Delta T = \left(\sum w_i * t_{task,avg}\right) * \left(1 - \frac{1}{N_{users}}\right) \qquad (4.3)$$

Equation 4.3 shows that the time savings is proportional to the number of users. This matches the findings of the NXConnect developers [1].

It is possible to demonstrate improvements in efficiency by examining the process used to define the tool paths for a part. Figure 5 shows an engine block that is machined from a solid billet. A similar engine block was

Figure 5. Engine block to be machined from Solid Billet. [9]

Table 1. Operations and user times for Engine Block shown in Fig. 5.

Operation	Weighting factor	User Time (Hours)
MOUNT BILLET UPRIGHT		
Face Front Face of Block	1.0	.5
Rough Mill Front Face	1.5	0.75
Semi Rough Front Face	2.0	1.0
Rough Main Hole Front Face	1.0	0.5
Drill 4 Holes Front Face	1.5	0.75
Drill 3 Holes Front Face	1.5	0.75
Bore Large Hole Front Face	1.75	0.875
Drill 12 Holes Front Face	2.0	1.0
Tap 12 Holes Front Face	2.0	1.0
Finish Front Face Square Corners	2.5	1.25
Finish Front Face Fillets	3.0	1.5
MOUNT FRONT FACE DOWN		
Face Back Face of Block	1.0	0.5
Rough Mill Back Face	1.5	0.75
Semi Rough Back Face	2.0	1.0
Drill 9 Holes Back Face	2.0	1.0
Tap 9 Holes Back Face	2.0	1.0
Drill 2 Holes Back Face	1.0	0.5
Counter Bore 2 Holes Back Face	1.5	0.75
Drill 2 Holes Back Face	1.0	0.5
Counter Bore 2 Holes Back Face	1.5	0.75
Drill 3 Holes Back Face	1.25	0.625
Counter Bore 3 Holes Back Face	1.5	0.75
Drill 5 Holes Back Face	1.5	0.75
Finish Front Face Square Corners	2.5	1.25
Finish Front Face Fillets	3.0	1.5
Bore Main Hole Back Face	2.5	1.25
MOUNT BILLET FLT IN 4 AXIS MILL		
Face Mill Slanted Faces	1.25	0.625
Face Mill Top Face	1.0	0.5
Rough Mill Top Cavity	2.5	1.25
Rough Mill Piston Cylinders	1.5	0.75
Rough Mill Sides of Block	1.5	0.75
Rough Bottom of Block	1.75	0.875
Semi Rough Top Cavity	2.0	1.0
Semi Rough Piston Cylinders	2.0	1.0
Semi Rough Sides of Block	2.0	1.0
Semi Rough Bottom of Block	2.0	1.0
Finish Top Cavity Square Corners	2.5	1.25
Finish Top Cavity Fillets 1	3.0	1.5
Finish Top Cavity Fillets 2	3.0	1.5
Finish Sides Square Corners	2.75	1.375
Finish Sides Fillets	2.75	1.375
Finish Cylinder Counter Bore	2.0	1.0
Finish Bottom Square Corners	2.5	1.25
Finish Bottom Fillets	3.0	1.5
Bore Cylinders to Finish	2.0	1.0
Drill 35 Holes Around Cylinders	3.0	1.25
Tap 35 Holes Around Cylinders	3.5	1.75
Drill 12 Holes in Top Cavity	2.5	1.25
Drill 1 Hole in Top Cavity	1.5	0.75
Counter Bore 1 Hole in Top Cavity	2.0	1.0
Drill 8 Holes Bottom of Block	2.0	1.0
Tap 8 Holes Bottom of Block	2.5	1.25
Total	-	51.75

found to require 100 hours of user time within a single user CAM application to define all the operations.

The details of the operations are enumerated in Tab. 1. Each operation is assigned an estimated weighting factor based on the time it takes to define the operation relative to the average time. The average time to define an operation for this part is assumed to be 0.5 hours. The user time is the time it takes for the user to define the operation, or the average time multiplied by the weighting factor. Note that each operation listed here represents a general task to be performed that may need several different operations to complete within the software.

Without including setup time and finish time, the time to define all operations by a single user is found to be 51.75 Hours.

Examining the operations it can be assumed that the tasks could easily be divided among five users. Using equation 4.3 and the results from the table we find that the time savings for this part and five users is 41.4 hours, or in other words, the group of five users would complete the work in just over 10 hours.

4.2. Expert users

One of the features of MUCAM is the ability to involve expert users. Often users are not experts in all manufacturing processes. Involving multiple users allows users who are experts in a process to develop the processes for that specific operation. The main benefit of this is that it reduces the likelihood of needing to redevelop 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.

When clients engage the model simultaneously there is a unique opportunity for experts to train novices in new capabilities. You can't get that with existing CAM applications, other than looking over the shoulder. Mixing expert users with non-expert users creates an ideal training environment. MUCAM allows effective knowledge transfer from expert users to novices.

4.3. 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. 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.

Central to the success of multi-user tool path planning 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 so can be assigned to individual users.

It was shown that the time savings is a function of the number of multi users. These estimates of time savings were found to be comparable to multi-user design 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.

Acknowledgements

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

ORCID

Andrew S. Priddis http://orcid.org/0000-0003-1549-3058
W. Edward Red http://orcid.org/0000-0002-9321-6913

References

[1] Hepworth, A.; Tew, K.; Trent, M.; Ricks, D.; Jensen, C.G.; Red, E.: Model consistency and conflict resolution with data preservation in multi-user computer aided design. J. Comput. Inf. Sci. Eng, 14(2), 2014. http://dx.doi.org/10.1115/1.4026553

[2] Marshall, F.: Model decomposition and constraints to parametrically partition design space in a collaborative CAx environment, Master's Thesis, Brigham Young University, 2011.

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

[4] NX Manufacturing Fundamentals, Siemens PLM Software Training Manual, 2007.

[5] Okulicz, K.: Virtual Reality-based approach to manufacturing process planning, International journal of production research, 42(17), 2004, 3493–3504. http://dx.doi.org/10.1080/00207540410001699426

[6] Red, E.; Holyoak, V.; Jensen, C.G.; Marshall, F.; Ryskamp, J.; Xu, Y.: v-cax: A research agenda for collaborative computer-aided applications, Computer-Aided Design and Applications, 7(3), 2010, 387–404. http://dx.doi.org/10.3722/cadaps.2010.387-404

[7] Red, E.; Jensen, C.G.; French, D.; Weerakoon, P.: Multi-user architectures for computer-aided engineering collaboration, Concurrent Enterprising (ICE), 17th International Conference on. IEEE, 2011.

[8] Red, E.; Jensen, G.; Weerakoon, P.; French, D.; Benzley, S.; Merkley, K.: Architectural limitations in multi-user computer-aided engineering applications, Computer & Information Science, 6(4), 2013. http://dx.doi.org/10.5539/cis.v6n4p1

[9] Sahov, J: Engine block, CAD model, https://grabcad.com/library/engine-block-16

[10] Xu, Y.: A flexible context architecture for a multi-user GUI, Master's thesis, Brigham Young University, 2011.

[11] Yau, H.; Menq, C.: Concurrent process planning for finish milling and dimensional inspection of sculptured surfaces in die and mould manufacturing, International Journal of Production Research, 31(11), 2709–2725, 1993. http://dx.doi.org/10.1080/00207549308956892