



2011-12-12

Model Decomposition and Constraints to Parametrically Partition Design Space in a Collaborative CAx Environment

Felicia Diane Marshall

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Mechanical Engineering Commons](#)

BYU ScholarsArchive Citation

Marshall, Felicia Diane, "Model Decomposition and Constraints to Parametrically Partition Design Space in a Collaborative CAx Environment" (2011). *All Theses and Dissertations*. 3184.

<https://scholarsarchive.byu.edu/etd/3184>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Model Decomposition and Constraints to Parametrically Partition
Design Space in a Collaborative CAx Environment

Felicia Marshall

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
Steven E. Benzley

Department of Mechanical Engineering
Brigham Young University
December 2011

Copyright © 2011 Felicia Marshall

All Rights Reserved

ABSTRACT

Model Decomposition and Constraints to Parametrically Partition Design Space in a Collaborative CAx Environment

Felicia Marshall
Department of Mechanical Engineering
Master of Science

An industry survey was conducted to collect information on current collaboration methods and project management and communication structures. The results, along with other design collaboration philosophies, were used to develop a method of coordinating users in a multi-user design space. These thesis methods will regulate collaboration and avoid user collisions in the same model space, either by cooperative interaction or by spatial decomposition with regional blocking.

The method partitions the design space by integrating a graphical user interface tool into the engineering application used to define and assign the necessary tasks of the project. A simple implementation of this method proved that it is usable by multiple users, is faster to setup than simple written instructions, and helps to coordinate users to work together efficiently.

To enable some of the key capabilities of the method, modern Computer-Aided application (CAx) architecture would need to be revised with multiple users in mind. One constraint example would be to partition the design space geometrically with visible boundaries between user-assigned areas. Current CAx architectures have some selection filtering capability that can be based on mathematical constraint boundaries, but are not designed to globally filter selection and are not very useful in their limited form. A simple solution to working around this limitation has not been found.

Keywords: decomposition, multi-user, GUI, CAD, partition, collaborative, coordination, constraints, boundaries

ACKNOWLEDGEMENTS

I would like to acknowledge all those who have helped me to complete this work, particularly my advisor, Dr. Red, and committee members, Dr. Jensen and Dr. Benzley as well as Vonn Holyoak and the rest of the v-CAx research group. I also appreciate the work of Jordan Ryskamp from which most of this research has evolved.

This work would never had been completed without the patience and support of my family and friends that helped take care of my daughter and home responsibilities to help me have the time I needed to work on this. I can't thank them enough.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	xi
1 Introduction.....	1
1.1 Problem Statement.....	2
1.2 Research Objectives.....	2
2 Background	5
2.1 Collaboration and Multi-User CAx	5
2.2 Constraints and Conflict Resolution.....	6
2.3 Decomposition Methods.....	7
2.4 Other Collaborative Environments	8
2.4.1 Online Gaming.....	9
2.4.2 Team Sports	10
2.4.3 Family Chores.....	10
3 Method	13
3.1 Survey of Industry Methods	13
3.1.1 Creating the Survey.....	14
3.1.2 Survey Respondents.....	14
3.1.3 Product Development Processes.....	15
3.1.4 CAx Tools.....	18
3.1.5 Desires for v-CAx	21
3.1.6 Concerns and Excitement	24
3.2 Collaboration Tool Design Criteria	26
3.2.1 Design Regions	28

3.2.2	Architecture.....	30
3.2.3	Features	31
3.2.4	Benefits	38
3.2.5	Limitations	39
3.3	Case Studies and Tool Evaluation	40
3.3.1	Case 1 - Engineering Design.....	41
3.3.2	Case 2 – Engineering Analysis	41
3.3.3	Case 3 – Code Development.....	42
4	Implementation	43
4.1	Design Criteria.....	43
4.1.1	NXCollab	44
4.1.2	Selection Filtering Tool	56
4.1.3	Benefits	59
4.1.4	Limitations	61
4.2	Tools Used	63
5	Conclusions and Recommendations	65
5.1	Conclusions.....	65
5.1.1	Architectural Changes.....	67
5.1.2	Tool Integration	68
5.1.3	Benefits	68
5.1.4	Drawbacks.....	69
5.2	Recommendations for Further Work	69
5.2.1	Visual Representation of Boundaries.....	70
5.2.2	Creation Boundaries.....	71
5.2.3	Tool Constraints.....	71

5.2.4	Complex Part Decomposition	71
5.2.5	Direct Conflict Resolution	72
5.2.6	PLM Applications	72
REFERENCES	75
Appendix A. Survey Results	79

LIST OF TABLES

Table 3-1: Ratings for Possible v-CAx Features	23
--	----

LIST OF FIGURES

Figure 3-1: Average Number of Employees Working on a Typical Complex Part.....	16
Figure 3-2: Max Number of Employees Working on a Typical Complex Part.....	16
Figure 3-3: Probability Companies Would add Employees to Work on a Part	17
Figure 3-4: Effectiveness of Communication Methods	17
Figure 3-5: Type of Organization at Each Company	18
Figure 3-6: CAD Programs Used by Companies of Respondents	19
Figure 3-7: Analysis Programs Used by Companies of Respondents	20
Figure 3-8: CAM Programs Used by Companies of Respondents	20
Figure 3-9: PLM Programs Used by Companies of Respondents	21
Figure 3-10: How Much Faster a new System Would Have to be for Acceptance	22
Figure 3-11: The Importance of Constraints to a Multiuser Engineering Environment.....	22
Figure 3-12: Example Design Region.....	29
Figure 3-13: Example of a GUI for the Task-Assigning Project Management Tool.....	31
Figure 3-14: Add Task Window	33
Figure 3-15: Add User Window	34
Figure 3-16: Add Group Window.....	34
Figure 3-17: Add Experience Level Window	35
Figure 3-18: Add Security Window.....	36
Figure 3-19: Add Order Window.....	37
Figure 3-20: Individual User Window Showing Tasks in Queue.....	37
Figure 4-1: Login Prompt	45
Figure 4-2: Import Tables Window	46
Figure 4-3: Create Task Manager Window	47

Figure 4-4: Add Restriction Window	48
Figure 4-5: Add Security Window.....	48
Figure 4-6: Add User Window	49
Figure 4-7: Add Group Window.....	50
Figure 4-8: Add Task Window	51
Figure 4-9: Add Order Window.....	52
Figure 4-10: Users Window for Editing	52
Figure 4-11: User Task List Window	53
Figure 4-12: Dependencies Window	54
Figure 4-13: Task Description Window	55
Figure 4-14: Selection Filtering Implementation.....	58

1 INTRODUCTION

Research is being conducted at BYU and in other research locations to develop a multi-user CAx environment for engineers to work simultaneously and collaboratively to expedite the design process. This is a major step because, “True innovation of the kind that drove the industry forward in the 70s and 80s seems to have died (CADAZZ 2004).” The Engineering Design Process, a serial system with multiple feedback cycles to develop engineering products, is less efficient in taking a product and the various components from concept to production than a parallel process could be. The question is raised: Is there a way to reduce the existing serial process to a collaborative parallel process? Research in collaborative engineering is answering the questions necessary to achieve that parallel process.

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. These constraints should preferably be administratively organized before designers begin collaboration or at least sometime during the project, wherever it becomes relevant. Companies, however, that are dependent on existing CAx tools are, according to Red, unlikely to “champion unconstrained (simultaneous) low-level model editing, because of the intense user interaction needed and the inconsistency of Internet communications (Red, et al. 2009).” Low level editing is where collaborators work simultaneously on the same part features, and in the same space,

causing interference and conflict, leading to intense interactions to resolve contextual misunderstandings.

1.1 Problem Statement

The objective of this thesis is to define and implement the constraints necessary for a collaborative CAx environment to be successful, while avoiding contextual interference where possible. A multi-user client-server collaborative software prototype, NXConnect, has already been developed (Ryskamp, et al. 2010). The API available in Siemens NX was used to extend the single user commercial CAx software to a multi-user environment in which users are able to concurrently create and edit a single model. Researchers have also worked to enhance the interface for this multi-user software to make better use of the collaborative potential (Xu 2010). This thesis extends previous research and adds to that multi-user prototype methods to partition the design space and constrain users to work within a specified design region.

1.2 Research Objectives

This research resolves many conflict issues among multiple users in a collaborative CAx environment. Specific research objectives follow:

1. Create a generalized method for model decomposition of single part files by developing administrative controls for parametrically dividing a model into tasks and user assigned regions.
2. Investigate and develop specific modeling constraints for multi-user CAx applications including geometric, feature-based, functional and order-specific constraints.

3. Devise a method for assigning users to specified tasks and regions and limiting their access and interactions with other regions of the model.
4. Determine a method for maintaining model continuity between user regions.
5. Demonstrate the effectiveness of the decomposition method in coordinating multiple users.

2 BACKGROUND

This work builds on the previous research of many other researchers, for example, Ryskamp, Lu, Lai and Sun. Their relevant contributions are described in these sections: 1) Collaboration and Multi-User CAx; 2) Constraints and Conflict Resolution; and 3) Decomposition Methods. The method also incorporates similar concepts from common virtual and non-virtual collaborative environments.

2.1 Collaboration and Multi-User CAx

Global competition and the advances of Internet technologies have sparked growing popularity for distributed collaborative applications that transcend “the traditional boundaries of physical and time zones (Fuh and Li 2004).” With this growing popularity, many have discovered the difficulties inherent to collaboration and have suggested some important attributes of a good collaborative system. “When group members are able to visualize and interact with each other’s datasets they are more likely to cooperate (Dempski, Harvey and Korytkowski n.d.)” 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.

Others have discussed the importance of distinguishing user roles. Lu, et al. observe that “while the technical decisions are dealing with ‘what’ and ‘how,’ the social interaction which is

about ‘why’ and ‘who’, is indispensable to the negotiations among the collaborative design decisions. (Lu and Jian Cai 2001)” A study done with children showed that the most effective collaboration in a shared digital environment occurred when each child assumed a territory with one acting as the “boss” (Olson, et al. 2011). Cera, et al. describe a method for hierarchical role-based viewing allowing users different level-of-detail (LOD) viewing based on the role of that user (Cera, et al. 2003). This helps the user to see how their portion fits into the whole without sharing unnecessary details, preventing distraction and protecting intellectual property. 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 and allowing for IP protection methods.

3D design applications present unique challenges that other multi-user applications do not because of “complex hierarchical and dependent relationships” between various objects in the environment (Agustina, et al., 2008). Other developments specifically related to CAD collaboration include methods to exchange data between CAD systems, maintaining consistency of design intent from one system to another (Sun, Ma and Huang 2009) using Hoffman’s Erep method (Hoffman and Juan 1993). The coordination method should thus be general enough to apply to any CAD system, though having a concurrent session between multiple CAD systems is not investigated in this research.

2.2 Constraints and Conflict Resolution

Constraint and conflict resolution methods are also relevant to this research. Lai, et al. explain “Geometric constraints are at the heart of computer-aided engineering applications (Lai 2009)” and goes on to say that “the ideal computer design tool for conceptual design is the one that feels natural and simple to use, rather than having sufficient power to handle anything you

can imagine.” The coordination method should be intuitive and simple requiring little set-up time to take away from the design process.

Many have suggested methods of locking or masking to govern the interaction of users. Bu, et al. explain a method for users to lock certain aspects of a design after they have worked on it including color, position, or everything about a feature and allowing annotations on the lock so other users can see their notes on the feature (Bu, Jiang and Chen 2006). Jing, et al. apply a local-locking concurrency control mechanism to lock features while a user is working on it, preventing other users from interfering with the changes being made (Jing, et al. 2009). The solution of Lin, et al. differs in that the users can all concurrently change features, but according to a priority schema only certain of the changes are displayed, though the other changes are stored in case higher priority changes are undone (Lin, et al. 2005). The method falls apart when constraints are applied dynamically, however. Similar to the priority schema, Chen, et al. created three coordination rules for conflict resolution in their e-Assembly to govern which user inputs took priority in situations of conflict (Chen, Song and Feng 2004). Synchronous technology overcomes the problem by changing from feature-tree dependencies to dynamic analysis of geometry (Gould 2008). This allows multiple users to make edits to a model simultaneously without the whole model having to update and re-execute features made after the one edited.

2.3 Decomposition Methods

Decomposition methods have been proposed as solutions to many different types of problems dealing with models already created. Chong, et al., describe a method for model decomposition and reduction to create non-manifold models for analysis faster than FEA methods (Chong, Kumar and Lee 2004). Cox, et al., establish a direct link between geometric modeling and continuum field modeling such that details automatically enter the analysis model

when added to the geometric model (Cox, Charlesworth and Anderson 1991). This could allow for geometry to be updated by the analysis model. Chan, et al. created a mathematical way to break up a rapid prototype model that is too big for the RP machine into smaller producible sections that can be assembled later (Chan and Tan 2005). Finally, Wei, et al. (Wei and Egbebu 2000) developed a technique for automatically generating all possible alternatives to machine a part, by “[partitioning] the design model into several useful smaller volumes which can be recognized as manufacturing features.”

Little has been done, though, in the area of decomposing design space before any modeling begins. Ram, et al., discuss the complexity of this idea, “the mechanisms of check-in and check-out of the design objects from shared space to local space of the designer appears to be an oversimplification of the collaboration needed in practical design environments (Ram, et al. 1997)”, later describing a better method: “Each designer works on his local design space occasionally interacting with design spaces located on other nodes.” It is not enough to just have the designers working in isolated volumes and try to put it all together later. “The system should support the user in knowing who is in the workspace, where they are working and what they are doing. (Sun, et al. 2006)” The coordination method should have several ways to decompose the design space allowing users to interact with each other when necessary and maintaining continuity between design spaces.

2.4 Other Collaborative Environments

People interact daily with many collaborative environments. Key insights can be gleaned from observing such environments. These sections discuss what can be applied to a virtual multi-user engineering environment from the following collaborative environments: 1) Online Gaming; 2) Team Sports; 3) Family Chores

2.4.1 Online Gaming

Massive multi-player online gaming has many parallels to virtual collaborative engineering. Because it is also a virtual environment, many of the concepts readily used in the gaming world can be directly applied to the engineering environment with similar architectural set-up.

First, in many games, users are constrained by experience level preventing them access to certain areas. This can be directly applied to the engineering environment constraining users by experience level, or can be adapted to constrain users by position in the company or function on the design team.

Second, users may have quests or tasks to complete specific to their user account. In the engineering environment, each user could have a list of tasks assigned to him that only he has access to. This list could contain individual tasks as well as tasks to be completed with the assistance of others.

Third, users see other's avatars and are encouraged to communicate via Voice-over IP or text-based chat. For a collaborative engineering environment, being able to see the work of others encourages communication and there has already been research done integrating VoIP and instant messaging with engineering tools (Mix, Jensen and Ryskamp 2010).

Finally, many games have users of different levels and skill-sets collaborate well together to complete group goals. Most engineering environments require a cross-functional team to create a single part or product and have them work together simultaneously to streamline the decision-making process.

2.4.2 Team Sports

Team sports provide a good example of a non-virtual collaborative environment that is well organized. The parallels are not as direct as with online gaming because of the translation to a virtual environment.

In team sports, just as in engineering project teams, participants are organized into teams of people with different skill sets and yet a unified goal. Those participants, though aware of the team objectives, are all aware of their individual responsibilities contributing to those goals. This should be the case with engineering teams as well, though sometimes a clear definition of each person's responsibility develops or evolves throughout the duration of the project.

Another major facet of team sports is that a referee regulates conflicts between participants. In some companies having a single decision-maker govern the conflicts could be an effective solution. This role could be that of the project team lead or manager.

2.4.3 Family Chores

Another useful environment to analyze is that of family chores. Again the parallels are less direct than the virtual collaborative environments, but the environment does provide some essential examples of the different types of overlap that can exist between participants.

Often in the family environment tasks are pre-assigned by a parent based on the overall objective of having a clean or functional home. Similarly, in an engineering environment, an authority figure of some type, assigns employees tasks based on the company's overall objectives. In the home the tasks are quite often assigned based on ability, as is also the case in engineering environments.

At home, as in engineering, some tasks are order-dependent while others overlap in space. The toys and clothes must be picked up from the floor before the person assigned to vacuum can successfully complete his task. Alternatively, you may have somebody trying to clean out the fridge while somebody else does the dishes while yet another person makes dinner, all in the same geometric space, without dependency. When creating a part in a computer-aided environment, many features are based off of previously defined features. Being aware of the dependency between the tasks of one user and another can greatly increase efficiency. Many features overlap in space as well, else Boolean operations would be of little use.

Finally, disputes are settled in a variety of ways in the home. Some can be handled by predetermined rules, others by compromise between disputants and others require the intervention of an authority, or parent. In an engineering environment, all of these and other conflict resolution methods are used. The collaboration method should have the flexibility to adapt to any one of these styles of conflict resolution.

3 METHOD

The collaboration tool described in this section is the result of integrating the important insights of previous researchers with the ideas from similar collaborative efforts discussed in Chapter 2. It focuses primarily on multi-user CAD environments but can be applied to any multi-user virtual environment of a technical nature. First, a survey of industry methods was conducted to understand current product development tools and procedures and to determine what features are viewed as most important to collaboration. The survey data was used to define criteria for the design of a collaboration tool to integrate with a development environment. The tool was then evaluated against several case studies and adjusted as necessary to create the general framework for creating a robust collaboration tool specific to the virtual environment and audience for which it is being created.

3.1 Survey of Industry Methods

A survey of industry methods was conducted with the objective of learning how product developers collaborate using current product development processes and single-user tools. The second objective of the survey was to extract their opinions on new multiuser applications, including the most important features and what constraints are necessary to facilitate

coordination efforts without impeding development progress. See Appendix A for the complete results.

3.1.1 Creating the Survey

The survey, specific to multi-user CAD, was created in two main parts corresponding with the two main objectives. The first questions were questions about the industry, size and distribution of the company and product development team. Then the questions focused more on the product development process: what types of parts were made, how long they took, the structure of the teams, the communication methods, the tools and applications used, and how they rated the collaboration of each of those applications. The second part was about the new multi-user tools: how much faster they would need to be, what features would be most important, what concerns they had, and what they were excited about. The features included in the list that was evaluated were generated by brainstorming ideas based on the background research discussed in Chapter 2.

3.1.2 Survey Respondents

The survey was sent out to over three hundred industry contacts in various engineering companies. Twenty-six of those responded by beginning the survey, while only fourteen completed the entire length of it. Those respondents came from companies varying in industry type from Aerospace and Automotive, to Education, and to CAx Software vending. They also varied in size from 500-5000 employees to greater than 15,000 employees. More than 70% of them came from companies with employees outside the U.S. and the other countries they employed varied from India to Mexico to Korea to Australia to Sweden and seemingly

everywhere in between. These facts agree with Fuh and Li showing the popularity for distributed teams and the need for distributed collaborative applications.

3.1.3 Product Development Processes

The first objective was to understand the product development processes and tools currently being used by industry. The questions addressed the difficulty of the parts made by each company, the resource allocation dedicated to the parts and the team communication and structure.

Examples of typical complex parts designed by the companies included engine block, combustion chamber, rocket motor nozzle, bladed disks, turbine vanes, car body sheet metal and bearing housings. The challenges faced by these companies in designing these complex parts included using multiple people to get loads, get geometry, perform part integration, build FE models, perform optimization, and certify components; integrating disciplines like Aerodynamics, Structures, Product Definition, Validation and others; balancing appeal and aesthetics with durability and safety, and designing for precision manufacturing where third and fourth decimal place tolerances are required. These types of large and complex parts are those most conducive to a constrained multi-user environment facilitating collaboration for product development.

Part development time varied from company to company but for most large companies such as these, the processes averaged more than 2 months. The number of employees dedicated to the development of the parts, however, showed a surprising split. As shown in Figure 3-1, the number of workers was usually between 2 and 10 or in the 100's (other).

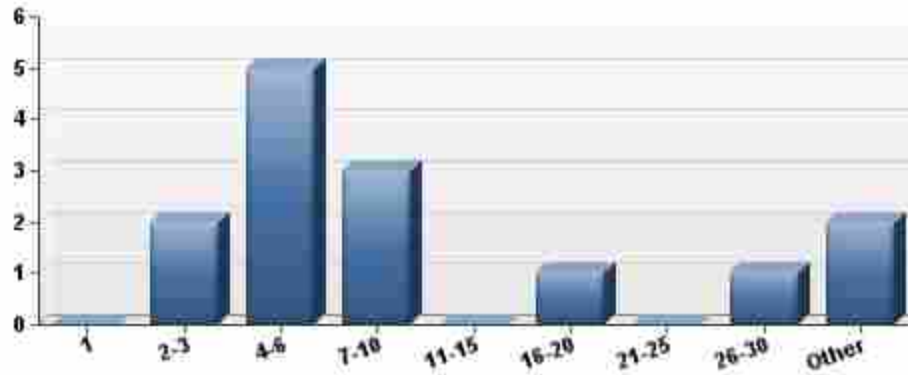


Figure 3-1: Average Number of Employees Working on a Typical Complex Part

When asked what the greatest number of people working on a single complex part would be, the ranges only moved up slightly as shown in Figure 3-2.



Figure 3-2: Max Number of Employees Working on a Typical Complex Part

Interestingly, most companies would consider adding more people to their projects if the work could be done efficiently. Figure 3-3 shows that only 2 respondents said they would not add additional employees to a project. This shows that companies want to apply resources to get projects completed faster but because of the limited capabilities of the single-user systems, it is not efficient to apply more people to a product design.

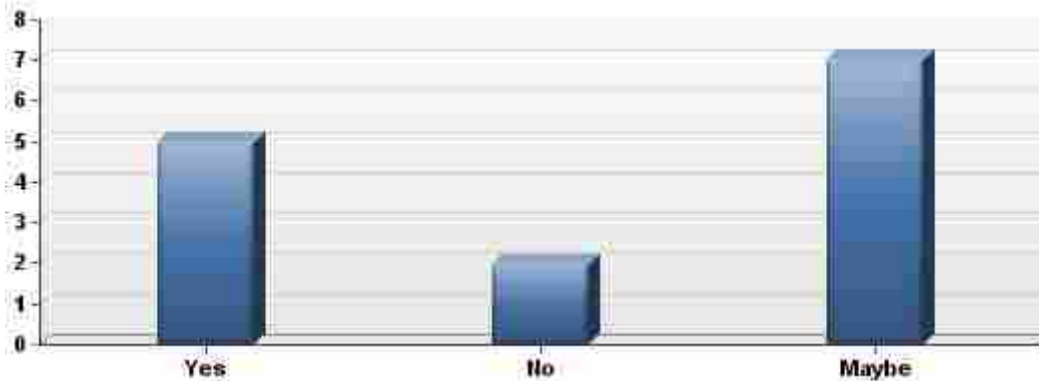


Figure 3-3: Probability Companies Would add Employees to Work on a Part

There are many different types of communication used by all companies with email and calls being the most popular and instant messaging being split between never and daily usage. Noteworthy is that these are all forms of contact that are not face-to-face. The industry is already accustomed to communicating through text and voice without visual cues such as integrating VoIP into the engineering tools (Mix, Jensen and Ryskamp 2010). Communication was usually rated as somewhat effective but varied from very ineffective to very effective (See Figure 3-4).

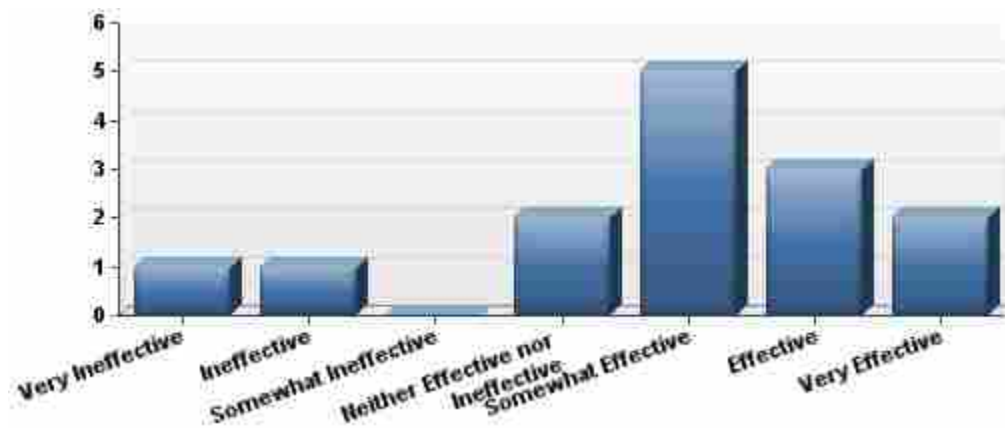


Figure 3-4: Effectiveness of Communication Methods

This demonstrates that there is always a need for better methods of communication. Putting users in a virtual environment together to see the developing work concurrently would certainly add clarity to the communication as Sun, et al. describes (Sun, et al. 2006).

Unsurprisingly, the most common project team organization is of a pyramidal nature as seen in Figure 3-5. Just as Olson, et al. showed with children, businesses have gravitated towards a structure with a single head, suggesting that a collaborative system should also adapt to the organization of an administrative role controlling the decomposition and constraint mechanisms in a multi-user environment (Olson, et al. 2011).

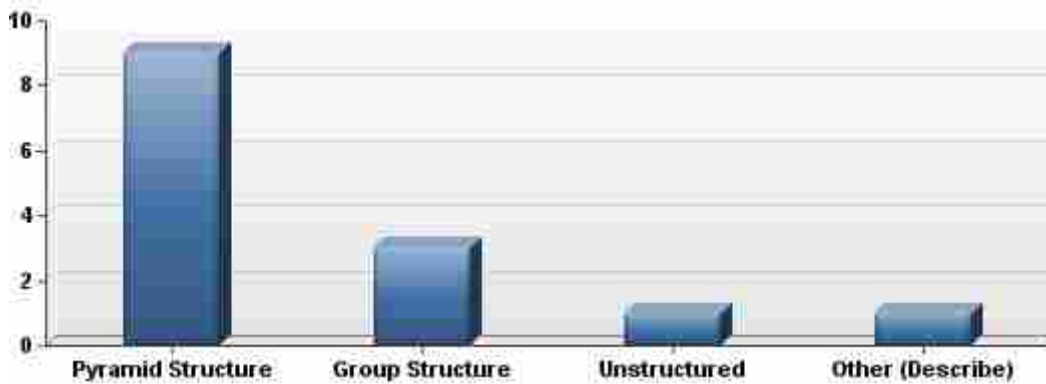


Figure 3-5: Type of Organization at Each Company

3.1.4 CAx Tools

To uncover what tools were being used at each company the types of engineering tools were divided into four categories: CAD, Analysis, CAM, and PLM tools. Respondents were also asked to comment on the collaboration capabilities of the applications they used, including what made them very good or very bad.

In the CAD category, every respondent indicated the use of several of the applications in their workplace. Figure 3-6 shows that every one of the 14 respondents utilized NX as well as others. The only comment about why CAD systems can be bad for collaboration is that their collaborative tools are not intuitive. This confirms Lai’s statement that “the ideal computer design tool for conceptual design is the one that feels natural and simple to use” (Lai 2009).

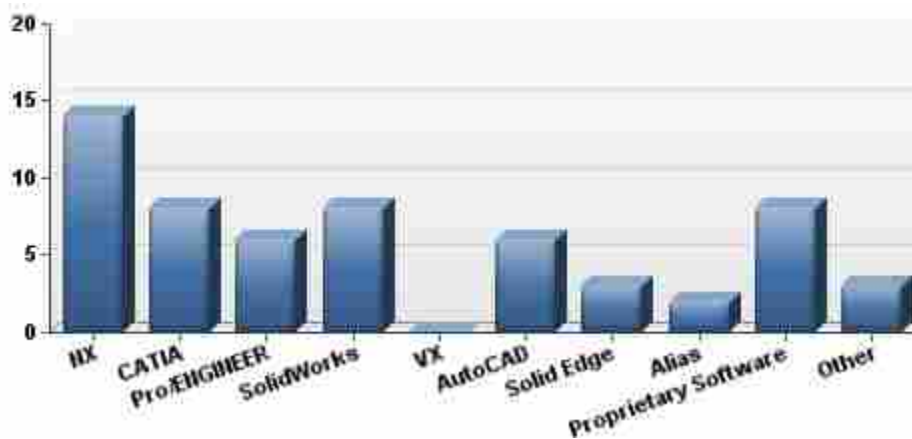


Figure 3-6: CAD Programs Used by Companies of Respondents

Analysis tools vary so much that every company uses several, often a different one for every type of analysis they do. Figure 3-7 shows that Ansys and NASTRAN were the most popular. Reasons for poor collaboration when it came to analysis packages included interfacing with loads and inputs, viewing, and real-time manipulation of the results. The only positive comment made was that some analysis tools have excellent file transfer capabilities for moving information between applications. Even this can be improved upon by maintaining design intent such as described by Sun, Ma and Huang (2009).

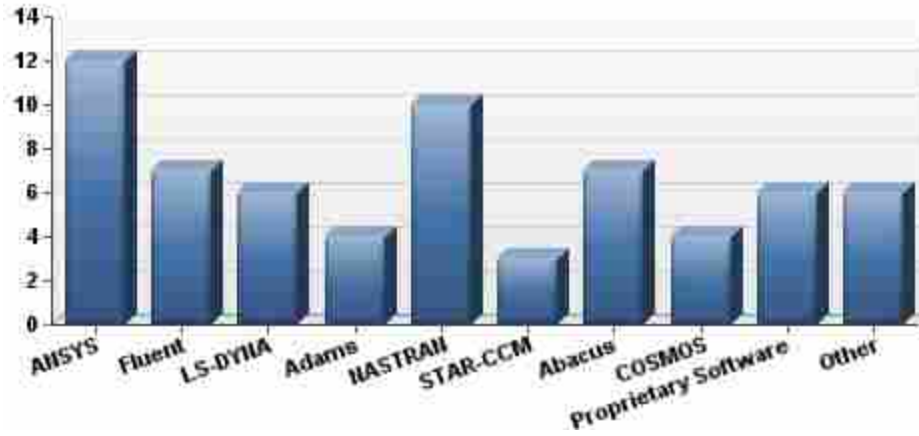


Figure 3-7: Analysis Programs Used by Companies of Respondents

For the manufacturing needs of the companies the independent CAM applications were not as popular as shown in Figure 3-8. Most companies used proprietary software or CAM tools that were built into the CAD programs (other) that they already used. It appears that industry appreciates integrated products that don't require further software and where that doesn't exist to meet their needs, it is common to create one. No comments were made concerning the quality of the collaboration capabilities of any of the programs.

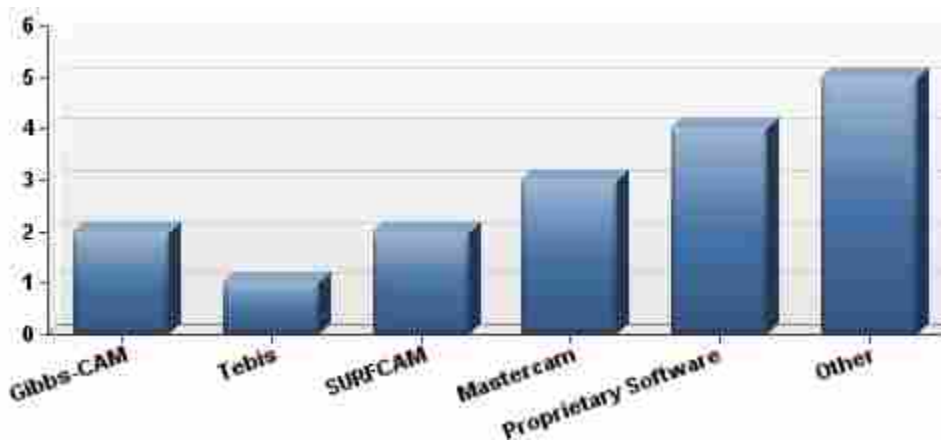


Figure 3-8: CAM Programs Used by Companies of Respondents

Finally, the look at PLM applications used by the companies of the respondents showed that the most popular was Teamcenter, as shown in Figure 3-9. In general companies thought Teamcenter to be good or very good in terms of collaboration. One comment noted that one reason a PLM application would be bad for collaboration is when the software is single decision point based and focused only on one user viewing or interacting with the files. This confirms the comments of Dempksi, Harvey and Korytkowski that “when group members are able to visualize and interact with each other’s datasets they are more likely to cooperate.”

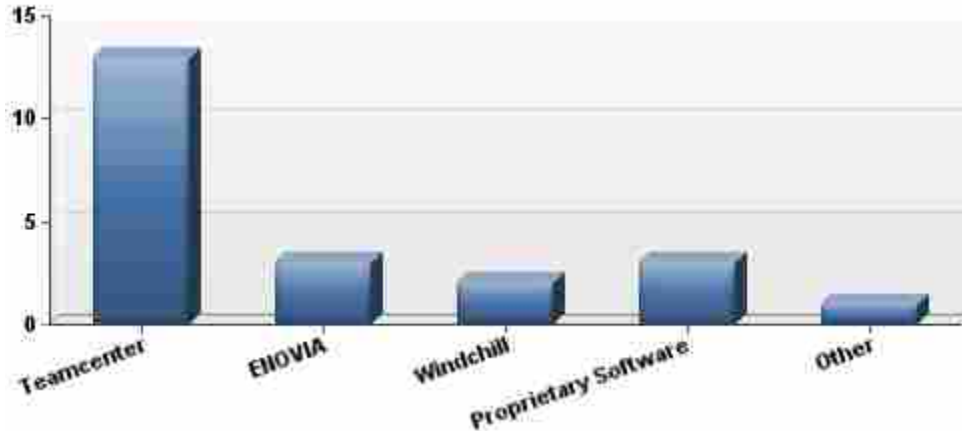


Figure 3-9: PLM Programs Used by Companies of Respondents

3.1.5 Desires for v-CAx

To understand better what the companies truly desired in a new collaborative engineering environment, the respondents were asked how much faster the system would have to be and what features would be the most important to include.

The results of asking respondents how much faster a new collaborative system would have to be to be accepted as a good investment for their company are shown in Figure 3-10. 64%

of respondents would be satisfied with a new multiuser system that made the product development process 50% faster than the current product development process. 100% would be satisfied if it were twice as fast as the current product development process.

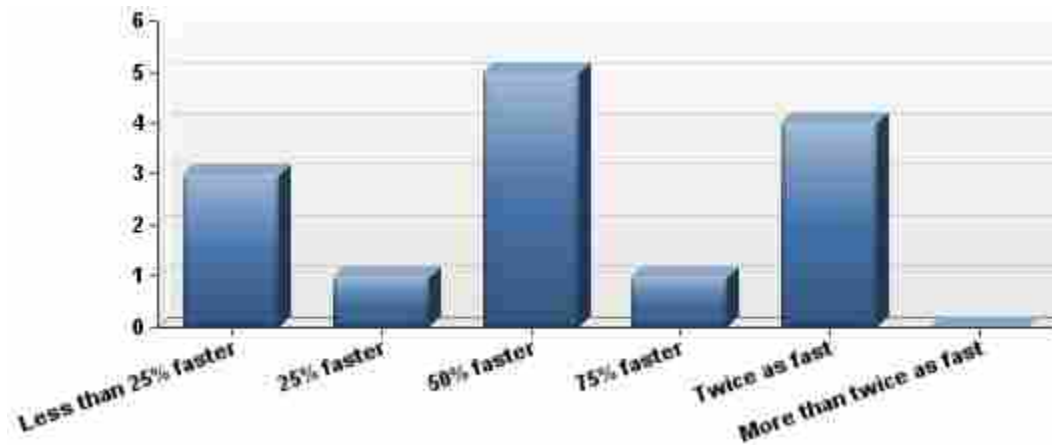


Figure 3-10: How Much Faster a new System Would Have to be for Acceptance

The results of asking respondents to rate the importance of constraints to a new collaborative engineering environment are shown in Figure 3-11.

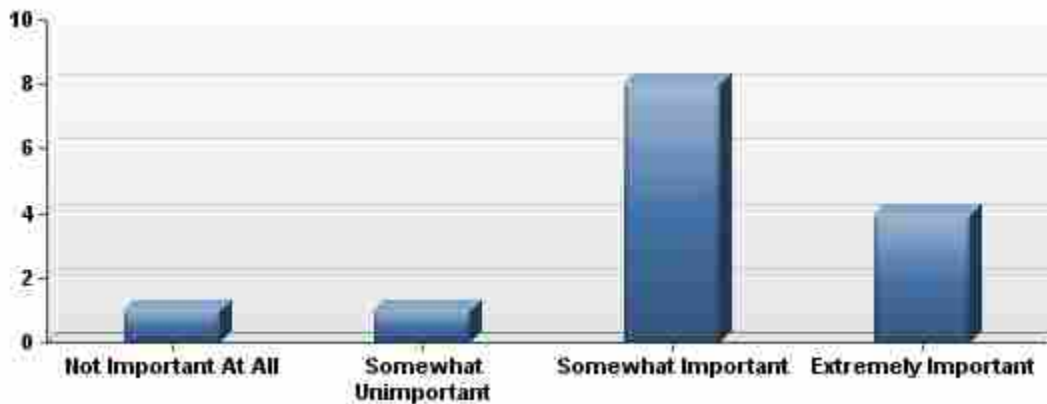


Figure 3-11: The Importance of Constraints to a Multiuser Engineering Environment

86% percent of respondents thought constraints on a multi-user system were important. 29% thought them to be extremely important. This data confirms the statement of Red, et al. that companies are unlikely to “champion unconstrained (simultaneous) low-level model editing (Red, et al. 2009).”

Finally, respondents were asked to rate a list of features that could be a part of a new multi-user CAD application. The listed items were results of the earlier mentioned collaborative research and are shown in Table 3-1. The ratings are as follows: Very useless (1), Useless (2), Somewhat useless (3), Somewhat useful (4), Useful (5), Very useful (6).

Table 3-1: Ratings for Possible v-CAx Features

<i>Method</i>	<i>Range of Responses</i>	<i>Most Popular Response</i>	<i>Average</i>
A user seeing a list of tasks each of his or her tasks are dependent on	2 to 6	5 (9 respondents)	4.93
A user seeing a list of tasks dependent on each of his or her tasks	2 to 6	5 (7 respondents)	4.93
A supervisory role allowed to grant permissions, or assign tasks to users	2 to 6	5 (10 respondents)	4.86
Listing tasks assigned to a user, whereby clicking on a task would take user to the area to work on that task	2 to 6	5 (8 respondents)	4.57
Allowing for users to lock geometric features from future editing with notes	2 to 6	5 (7 respondents)	4.50
Requiring permission from original creator to delete or edit a feature	2 to 6	5 (8 respondents)	4.50
Allowing one or many users to be assigned to a single task such as a sketch	2 to 6	5 (7 respondents)	4.50
Locking certain tools from the use of one or more users	2 to 6	5 (5 respondents)	4.21
Permission to work or view only when granted entrance to another user's area	2 to 6	3,4,5,6 (3 respondents)	4.14
Having details of other users' tasks require permissions to be viewed	2 to 6	2,3,4,5 (3 respondents)	3.86
Confining users to work within a predetermined geometric space	2 to 6	3,4 (4 respondents)	3.79

All of the ideas for constraint features were rated highly, showing that any and all user-constraint capabilities will be appreciated. Important features to include as part of the constraints for a multiuser engineering environment are those top rated items in the table:

- A user seeing a list of tasks each of his or her tasks are dependent on.
- A user seeing a list of tasks dependent on each of his or her tasks.
- A supervisory role where an administrator is allowed to grant permissions, or assign tasks to users.

These are top-rated items again align with the research of Olson, et al. in that users need to be aware of their territory and a single user should be responsible for defining the team interactions. Locking features with notes as suggested by Bu, Jiang and Chen also received a high rating. The declaration of Ram, et al. that “the mechanisms of check-in and check-out of the design objects from shared space to local space of the designer appear to be an oversimplification of the collaboration needed in practical design environments” explains the relatively lower rating of confining users to work within a predetermined geometric space. The rating itself, however, is still quite high, showing that many users see the potential benefits of the geometrical constraints as long as the interaction between design spaces is not completely eliminated.

3.1.6 Concerns and Excitement

The survey concluded with a comments area to describe the concerns and the excitements the respondents had after working through the survey and learning about the ongoing research related to the multiuser environment. Some valuable insights were gained from looking through the responses.

Some of the concerns were related to security, speed and scalability of the entire system and not specifically related to collaboration constraints; however, on a smaller scale, the security,

speed and scalability of the collaboration tool is essential. Other concerns were more focused on the collaboration aspect. One respondent said,

“While some of the access control ideas above are interesting, the nature of rapid development may make these less desirable in practice. When we're really trying to move fast on a project, I envision us wanting our experts to have *more* ability to fluidly move from area to area, helping as needed, rather than less. I could see the access controls quickly getting in the way. Concepts around tracking decisions and associating decision-making with geometry driven by those decisions will be increasingly important in this world. It will be extremely easy for a designer who is new to a team project to step in and start second-guessing the cumulative wisdom of the team to date -- particularly if there is not a very simple way for that new designer to understand which aspects of the design were very deliberately determined, and which are more open for modification.”

Essentially the method needs to be flexible enough for those who want heavy controls and those who want few controls to all be able to use it effectively. It would also be helpful to be able to add notes about the design to the features for others to see as desired. This capture of design intent and awareness of the “why” behind the “who” has been noted as important by the research of Sun, Ma and Huang; Bu, Jiang and Chen; and Lu, et al. Notes could also be used to help establish design requirements, another concern. The tool needs to integrate well with the system and the design space and workflow needs to be well-managed and efficient, to resolve other concerns.

On the other end, there were many things learned about what the expectations of the collaboration tool are from the positive comments. Some were just hoping that the research would motivate commercial tool vendors to integrate more collaboration tools. Others were excited at the possibility of company personnel and customers working together or multiple experts in a company “ganging up on” a complex model to speed up the design process and reduce scrap. Another comment highlighted the increased ability to change workload scheduling, people allocations, priorities and hand-offs to focus on areas needing more work for analyses. And finally, one respondent mentioned it would be a great mentoring tool to allow a

“senior” designer to be in the same environment overseeing a “junior” designer. These ideas correlate well with the research of Cera, et al. describing role-based viewing. While none of the comments were specifically related to the constraint requirements, these expected benefits must be met and flourish with the help of the added collaboration tool.

3.2 Collaboration Tool Design Criteria

The design of the collaboration tool of the multi-user environment considers both the research described in Chapter 2 and feedback from the industry survey. Catering to the requirements and desires of those that use the software will make them advocates for it, but scientific research in the field often highlights other insights unforeseen by regular end-users, who are often limited in vision by currently used tools.

From the literature review, the following design criteria for the tool were determined:

- Don't completely isolate users from each other
- Allow users to see developing work of others and encourage communication.
- Distinguish between roles of users
- Give each user ownership of a portion of the design space
- Allow for IP protection methods
- Be intuitive and simple, requiring little set-up time
- Have several ways to decompose the design space
- Allow users to interact with each other when necessary
- Maintain continuity between design spaces

Reviewing common collaborative environments provided additional design criteria:

- Have option to constrain users by experience level or function on the team

- Give users list of assigned tasks – individual and group tasks
- Allow for a cross-functional team to work together simultaneously
- Make users aware of individual responsibilities contributing to team goals
- Allow a single decision-maker to govern conflicts
- Allow an authority figure to assign tasks perhaps based on ability
- Help users to be aware of dependency between tasks and spacial overlap
- Have flexibility to adapt to multiple styles of conflict resolution

From the survey results, more design criteria was outlined:

- Allow interactions and decision-making by multiple people
- Include as many constraint capabilities as possible
- Allow a user to see a list of tasks his or her tasks are dependent on
- Allow a user to see a list of tasks dependent on each of his or her tasks
- Allow an administrator to grant permissions or assign tasks to users
- Flexible to allow for heavy controls or few controls
- Allow for the addition of notes in association with features
- Be integrated with the system and design space
- Manage workflow well and efficiently
- Allow for users with limited abilities mainly used for viewing
- Allow for multiple users to work in same space on same task
- Allow for regular changing of tasks assignments and priorities

The collaboration tool was designed around these criteria. Thus it provides necessary coordination capabilities and constraints while maintaining enough flexibility to adapt to the

different styles of collaboration and various work environments described by the survey respondents.

This research combines the recommendations and findings of many researchers in the areas of collaboration, part decomposition and conflict resolution, with observations of paralleling collaborative environments and industry insights and opinions. In so doing, it provides a solid framework for designing a collaborative tool specific to any multi-user application. This research is also the first to investigate the architectural requirements for implementing user-constraints as part of the collaborative tool, pointing out limitations of current single-user architectures.

3.2.1 Design Regions

The difficulty in decomposing design space for multiple users to work in is that there are so many ways to divide up the space. Beyond the geometric decomposition of Chan and Tan for rapid prototyping (Chan and Tan 2005), or Wei and Egbelu for machining (Wei and Egbelu 2000), design volumes can be overlapping and model features can be additive or subtractive. It is in these situations where it is imperative that the design regions are clearly defined. In a CAx environment, a model has traditionally been made up of a series of features defined and displayed in a feature tree. The decomposition is therefore most easily decomposed into its features rather than strictly by its geometry. Further difficulty is added when breaking up a model into features or groups of features (tasks) before they are created – dividing empty space, or in breaking up a part into time-dependent features where one cannot be accomplished until the other is completed. Consider a part, P made up of regions $A-L$, where

$$P \equiv \{A, B, C, D, E, F, G, H, I, J, K, L\}. \quad (3-1)$$

Each user would then be assigned to a subset of P but those subsets would not necessarily be exclusive (see Figure 3-12):

$$\text{User 1: } P1 \equiv \{A, B\}, P1 \subset P \quad (3-2)$$

$$\text{User 2: } P2 \equiv \{B, D\}, P2 \subset P \quad (3-3)$$

$$\text{User 3: } P3 \equiv \{C\}, P3 \subset P \quad (3-4)$$

$$\text{User 4: } P4 \equiv \{E\}, P4 \subset P \quad (3-5)$$

$$\text{User 5: } P5 \equiv \{F, G, H, I\}, P5 \subset P \quad (3-6)$$

$$\text{User 6: } P6 \equiv \{H, K\}, P6 \subset P \quad (3-7)$$

$$\text{User 7: } P7 \equiv \{E, G, H, I, J, K, L\}, P7 \subset P \quad (3-8)$$

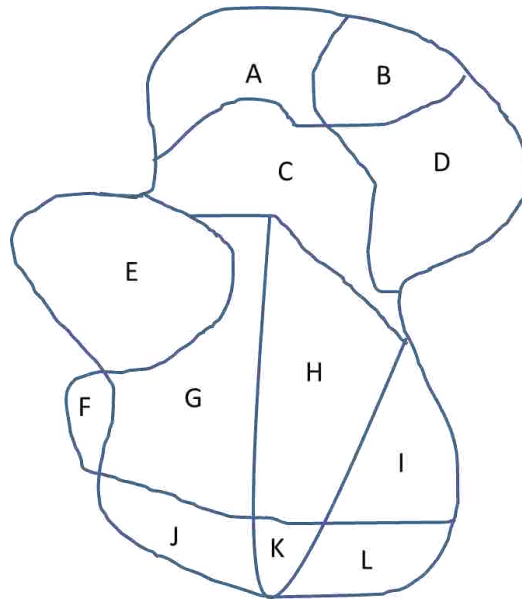


Figure 3-12: Example Design Region

In this example User 3 is the only one that does not need to collaborate with any other users. Regions A, D, F, J, and L are all exclusively owned, but the users assigned to them, have more regions in their assignment that they must collaborate with other users on. Region H has

three users assigned to it. The following method assumes that design regions are assigned exclusively or that collaboration between users in overlapping regions can be successful through adequate communication rather than by functional constraints. Further research into direct conflict resolution may provide a future addition to the method that removes the need for this assumption (see Chapter 5).

3.2.2 Architecture

Because of the need for an administrative role and task assignment, the coordination tool is structured as an organizational tool. The basic idea is that one or more administrator(s) or project leader(s) would set up the project using a graphical user interface for task definition and assignment. Members of the team contributing to the project would then use the tool to track the progress of their own and others' tasks.

The architecture consists of that graphical user interface connecting to a database of separate but interconnected tables: one for tasks, one for users, one for groups, one for security ranks, one for restrictions, and generally, one for any information to be stored in association with the tasks or users. The database is stored on the same server as the application data that is being shared. In order to integrate with the system and design space, the GUI itself should be integrated with the application such that it blends into the design and architecture. To make it more transparent, the tool should be customized to include features or terminology specific to that application.

In order to implement many of the desired constraint features, the underlying architecture of the application needs to consider the interaction of multiple users and allow for features or spaces of functions to be filtered and limited based on the user. A constraint filter would have to approve or reject all intended actions such as creating, editing, selecting and viewing by

comparing the intended action against the user or task based restrictions. Integrating this filter is the most difficult aspect of the architecture because it is specific to multiple-users and is not easy to layer on top of single-user applications.

3.2.3 Features

The GUI that the administrator(s) or project leader(s) would use should allow for the creation and editing of tasks, users, dependencies, and any other component of the customized tool (see Figure 3-13). It should also display information in a quick and easy to understand format, such as a table structure.

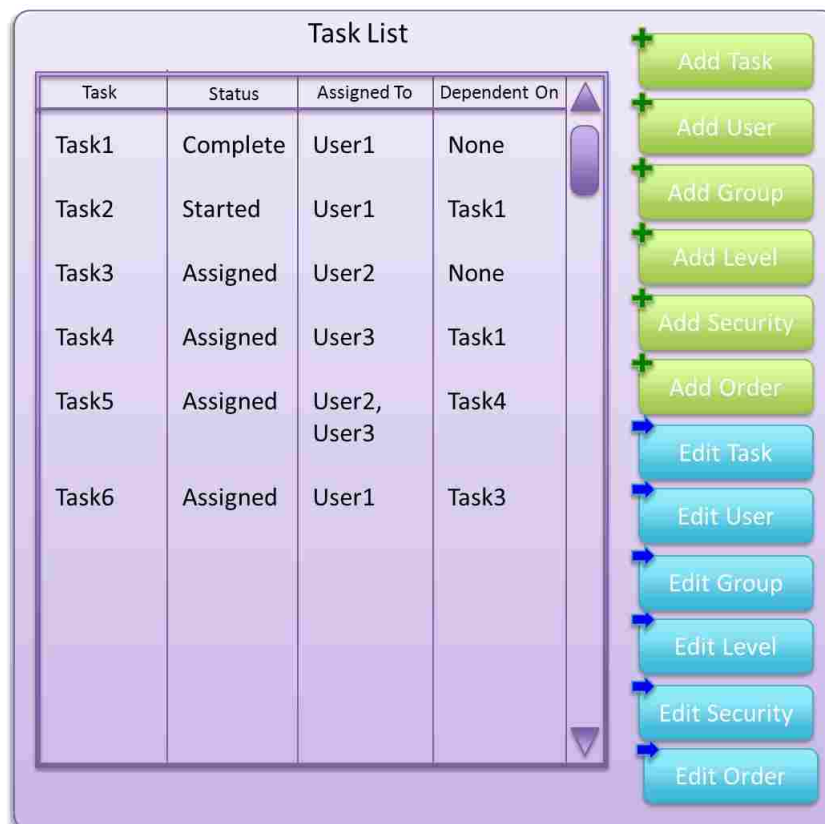


Figure 3-13: Example of a GUI for the Task-Assigning Project Management Tool

Depending on the application for which the tool is created, the details shown in this table could vary. For instance, many engineering companies would prefer to display status as a percentage of work completed. Other companies may find a due date or priority to be more crucial information. For this reason, the task list should be designed to the application and as customizable as reasonable.

Tasks should include anything useful to the user or administrator specifically related to the task such as the following options: name, description of the task, the option to direct the user to a specific area to work, dependence on other tasks, priority, a required clearance level, a list of users it is assigned to, visibility options, working area restrictions and a notes section for listing anything related to the task that may be common across tasks or other useful information. The definition of a task is anything that can be assigned to a user or group of users. This allows for as much or as little detail as desired by the administrator. For example, one company may have very specific details and extensive constraints to associate with a particular task of the product development while another company may have very general descriptions of tasks to be interpreted and defined more specifically by the assignee.

All of these task characteristics should be defined as part of the task definition window, such as that shown in Figure 3-14. Each should be customized to the application and the type of information that would be important to each task. For example, there could be a work area designation. For engineering models, the administrator could define geometrically and parametrically what area of the model is the work area for that task. The area could be defined mathematically by the administrator inputting equations, or graphically by selecting existing planes and surfaces and defining the allowable side, much like defining a sketching plane and viewing direction. The user assigned to the task would then not be able to select anything outside

of the work area while working on the task. The area restrictions could be completely different for a different task. For programming applications the boundaries could be certain lines of code or certain files.

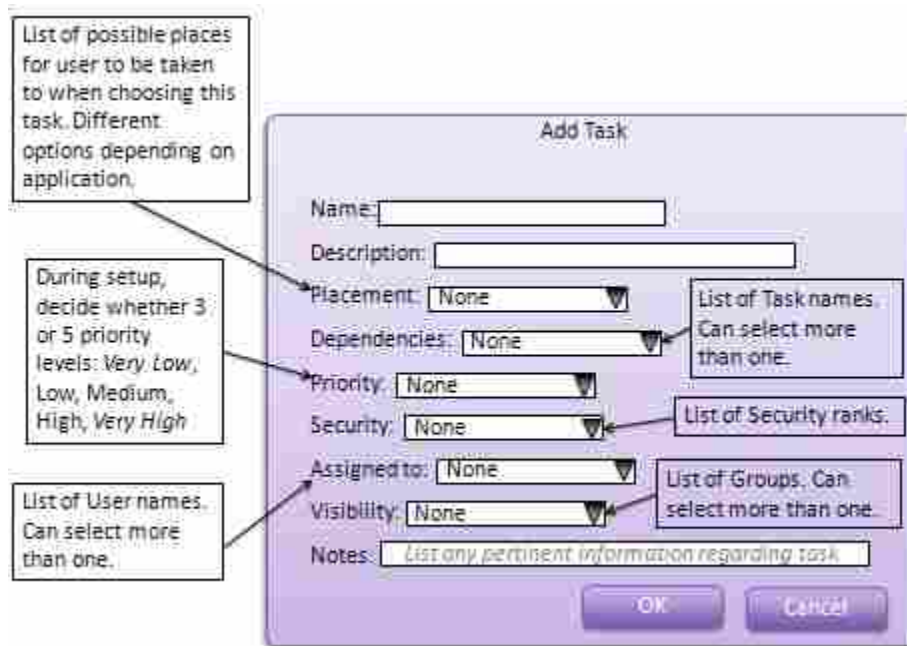


Figure 3-14: Add Task Window

Users should also have characteristics associated with them. They could have a group they are a part of, a security rank associated with them, tool restrictions associated with them, and an indication of what skills a user may have (See Figure 3-15).

All users that have been added to a project should be able to add tasks at any time during the project as they see needs arise, but limited by administrators who retain certain exclusive controls such as priority or task assignment. The administrator could edit tasks to add those more sensitive pieces of information or delete a task if deemed unnecessary. Users could also have a security rank associated with them allowing them to only be assigned to create tasks with a

security rank equal to or less than their own. Administrators, however, should be able to change the rank of users or tasks to be higher or lower.

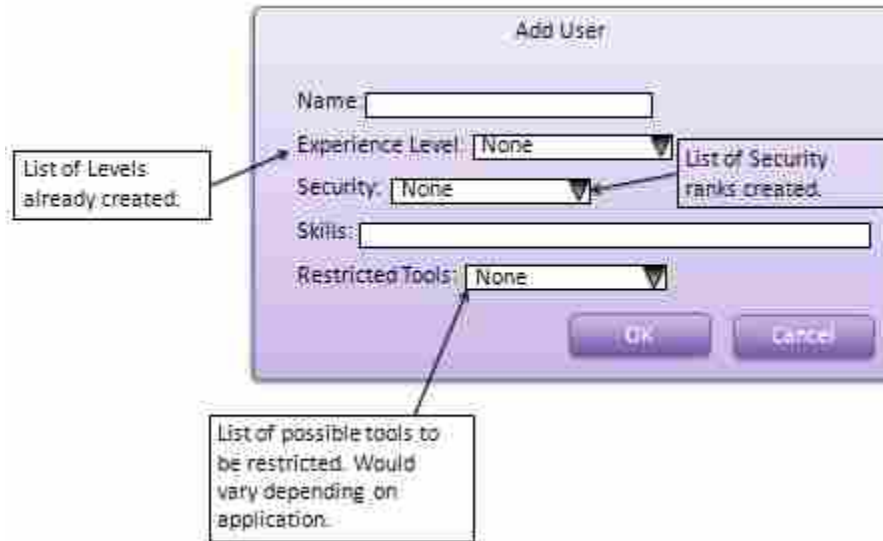


Figure 3-15: Add User Window

A group could be made to group users together much like an email list group (See Figure 3-16). When creating a task, it can be made visible to only certain groups. Each group could be given tool or geometric restrictions and any user could be made a member of any or multiple groups.



Figure 3-16: Add Group Window

As part of the task definition, user definition, or other delineating feature definition windows (such as groups or ranks), the administrator should be able to restrict certain tools from use by the user(s). If associated with a task, the restriction would apply only while working on the task. If a restriction is associated with an experience level or security rank, then the restriction would apply until a higher level or rank was achieved (See Figure 3-17). If associated with a user or group, the restriction would apply at all times until an administrator removed it.

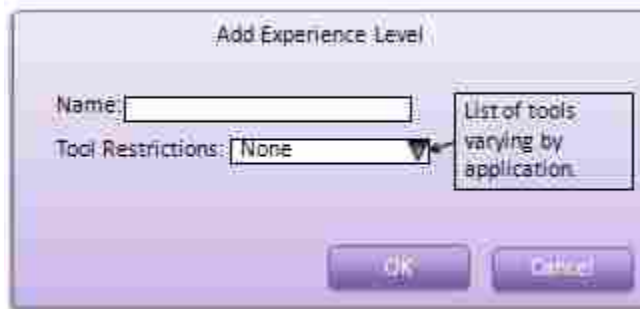


Figure 3-17: Add Experience Level Window

The option should exist to restrict each user's selecting and/or viewing capabilities. Users might only be allowed to select or view features that have a security level equal to or less than their own (See Figure 3-18). The definition is again left up to the company or administrator as to what security levels they would use. Tasks might also have a visibility characteristic assigned to it, allowing only certain users or groups to see the features produced as part of the task. Users could also be limited in what features they could select and what tools they could use while completing a task. Users might also have general restrictions with defined boundaries that constantly confine their selection capabilities to a specified volume of the design space.

Users could be allowed to only work on tasks that are assigned to them or for which they have permissions to join. Permissions could be granted by the owner of the task or an

administrator. Permissions could be restricted further by the security level of the user requesting permissions, allowing joining permission to only be granted in the case that the security level of the user is equal to or greater than that of the task. Granting such permissions would give the permissible user editing capabilities or perhaps only viewing rights. Restrictions such as these are not issues encountered in a single-user environment and would require that the architecture was planned with capabilities to filter actions based on user or task specific restrictions and permissions.



Figure 3-18: Add Security Window

Order-specific constraints are inherent to dependency based task availability. Tasks that depend on the completion of other tasks may only be selected once the tasks they are dependent on have been designated as completed by the owner of that task (See Figure 3-19). An example of this would be putting holes into a block; The holes cannot be made until the block has been made. In such a case, the task of creating the holes should be designated as dependent on the task to create the block. The user assigned to create the holes would then be prohibited from selecting

that task to work on it until the user assigned to create the block had completed that work and designated its status as completed.

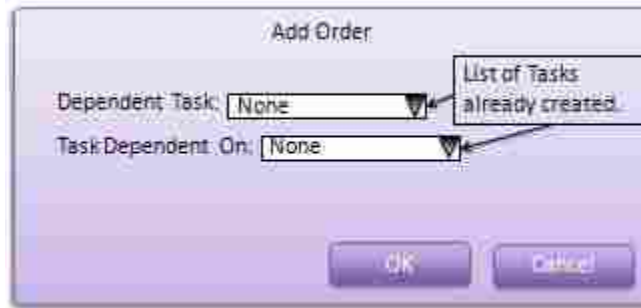


Figure 3-19: Add Order Window

Once assignments have been made, each user has a window listing the tasks assigned to him (See Figure 3-20).

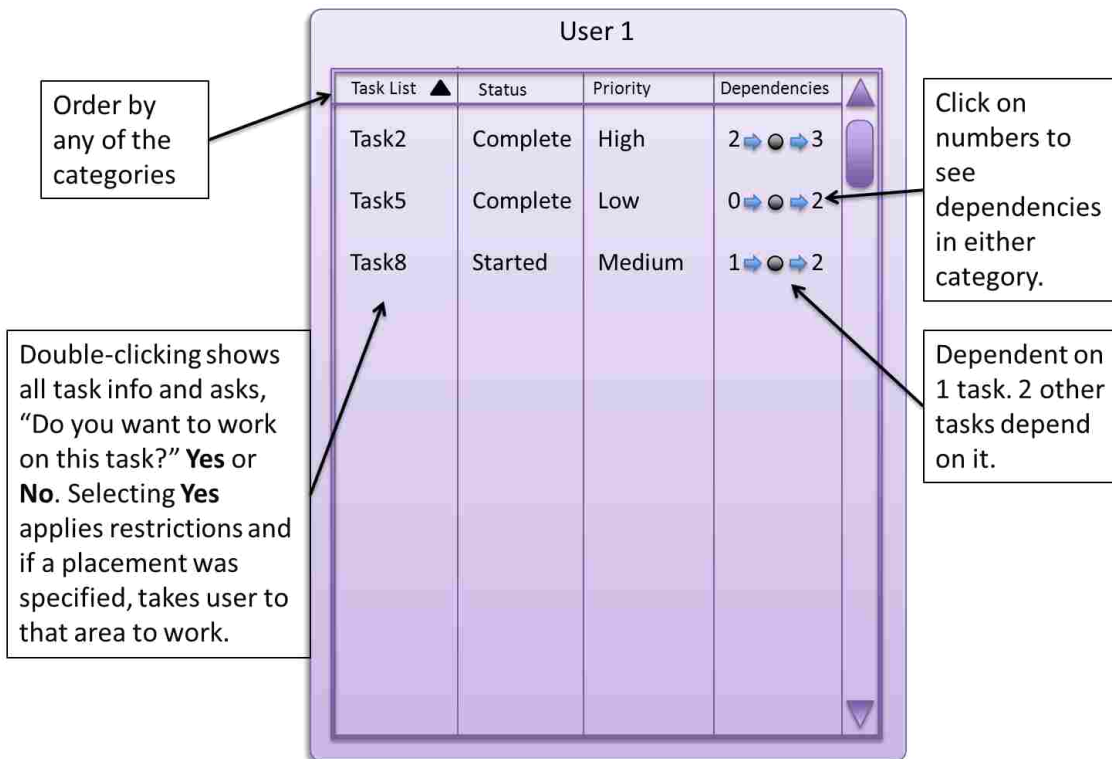


Figure 3-20: Individual User Window Showing Tasks in Queue

The dependencies, designated priority, and status of each task or other key characteristics of the task should also be listed. Double-clicking on a task could take the user to the designated area to work on the task if an area was defined. Users could be prevented from working on anything until they have selected a task to work on.

3.2.4 Benefits

The benefits of using a collaboration tool such as has been described are extensive. Feature-based constraints are inherent to the task-based assignment method. By creating tasks, an administrator can define which features should be made by what user in what area and in what order. Users can only work on tasks they are assigned to, providing focus and direction to users. Users can't work at all unless they have selected a task from their list, eliminating aimless and chaotic editing. Users may be invited to join the task of another user, such that assistance could be provided where needed. Users may also seek permission to join another task, by asking the user assigned to the task or asking the task administrator. This way a user waiting on dependent tasks to be completed could join the efforts of those they are assigned to and speed up the design process, while eliminating unproductive time.

This tool is defined such that tasks could also be annotated or locked by the creator with notes describing the design decisions, similar to the locking mechanisms described by Jing, et al. (Jing, et al. 2009). For this tool, however, the features of the task would be locked for editing, not only while the user was working on the task, but even after the user had completed it. It would then require permission from the creator or an administrator to override. Again, this locking mechanism would have to be built into the architecture such that a filter could detect that the features were locked and prevent others from editing them.

The biggest benefit of the method is that the project elements still interact with one another as they would in a single-user environment. It would not interfere with the “complex hierarchical and dependent relationships” between various objects in the environment (Agustina, et al., 2008). Thus, it should integrate well with the applications it is applied to because it doesn't affect the core interactions of the features. In other applications, like that of Synchronous Technology's dynamic analysis of geometry, the method could be applied to work just as well.

Tasks would also be separated in a fashion suitable to working in an environment with limited viewing or selecting such that geometry located at boundaries between regions would be continuous. For instance, in making a wheel, the task assigned to make the spokes could be bounded by the outer diameter of the hub and the inner diameter of the wheel rim. If either would diameter changed, so would the boundary restrictions on the spoke task, adjusting parametrically. Basing several features on the same parameters or variables will ensure seamless boundaries between regions.

3.2.5 Limitations

This tool is specifically designed to coordinate the efforts of multiple users on complex parts where weeks of time or more are required, rather than days. For less complex models the setup time required for the tool to be of any use at all would not be worth the effort. Even for the most complex parts, the setup time may prove to be a hindrance to this method. It is possible that teams will not want to use such a detailed and systematic approach to design. It requires forethought on the part of the administrators or project leads to break apart the project into manageable and assignable tasks. Fortunately, it is flexible enough to allow any definition of manageable and assignable. Research should be done in developing a systematic approach to breaking apart the project into tasks to more efficiently utilize this powerful tool (See Chapter 5).

Contrary to the research of Jing, et al; Lin, et al; and Chen, Song and Feng, this method attempts to prevent direct conflict resolution by encouraging communication and understanding between users working together or near one another, and by constraining user interactions, rather than defining the application response in situations of conflict. The problem of direct conflict resolution is complex and requires further research, particularly in the context of multi-user applications (See Chapter 5).

The architecture of single-user applications to which this method would be applied may have to be redesigned to consider the interaction of multiple-users and the constraint features that would be necessary for that environment. Many engineering applications have architectures conducive only to a single user and a single screen. Adding the dynamic of multiple users with complex interaction constraints could require extensive workarounds to fit on top of the current architectures. The application should instead be re-architected with multiple-users and interaction constraints at the core of the design. This also requires extensive work and access to the source code of the application. It would require that the developers of the application create the multi-user version of the application rather than a third party creating an add-on for it.

3.3 Case Studies and Tool Evaluation

The collaboration tool was evaluated against three application cases: (1) Engineering Design, (2) Engineering Analysis and (3) Code Development. The multi-user tool specifications and design features were adjusted as necessary to more generally adapt to the varied situations.

3.3.1 Case 1 - Engineering Design

Engineering design is the application around which the tool was created so it was pertinent that the tool work well in this application. For this reason, the initial design was created with the principles of engineering design at the forefront. Thus, when running through the method with engineering design as the application (CAD, or Computer-Aided Design), the tool design was left almost unaltered. One addition as a result of this case study was the tool restrictions applied to specific users or tasks.

3.3.2 Case 2 – Engineering Analysis

The engineering analysis application (CAA, Computer-Aided Analysis) proved to be very similar to the engineering design application. The essential difference between the two in the context of this tool design is that with analysis users would often be working on different tasks on the same features of the part. Ideally one user could be performing one type of analysis of a certain area of the part while another was working on a different type of analysis of the same area.

When multiple users are working in the same area on very different tasks, the ability to hide certain features from certain types of users becomes essential, not only for security but also to guard a user from the unnecessary distractions of the work of other users appearing. Thus the change was made that in addition to security ranks, there would be work group visibility options, allowing for the features of some tasks to be made visible only to certain work groups.

3.3.3 Case 3 – Code Development

The case study for code development proved to be the most altering to the general tool design. Code development is different enough from both engineering design and analysis that the method had to be more generalized to encompass it.

An example of one such generalization is the placement option, on the Add Task menu. This option was originally dedicated to a specific plane or surface that a user could be directed to upon selection of the task. This would be associated with tasks like sketching or creating a feature on a surface which would not apply at all to code development. Thus the placement option would vary based on the application to allow for the selection of a plane or surface for engineering design or analysis scenarios while also allowing for the selection of a particular line in the code or a certain file for development scenarios. In this generalization, all applications could program a task to take the user directly to the affected area upon selection.

Terminology was also adjusted as part of this case study. Particularly, “Clearance Rank” was changed to “Security” and “Parameters” was changed to “Notes”. The functions of these fields didn’t change, but the term security is likely better understood by a wide variety of applications, and the term notes is most certainly more widely understood than parameters.

4 IMPLEMENTATION

To demonstrate the method, a simplified implementation of many of the important aspects of the tool design was developed. Because the method is generalized, any testing prototypes would be specialized. The general method was applied to the case of engineering design, or multi-user CAD. Following the method, the survey of industry methods and past research were used to propose the design criteria for the implementation of the collaboration tool. Then the necessary features for the multi-user prototype were developed and tested building on the current architectures of the CAD system being used.

4.1 Design Criteria

The list of design criteria from the research and industry survey was extensive. However, for the purpose of the implementation, it was unrealistic to incorporate all the listed features. Instead the focus was placed on those features most highly rated by the survey respondents.

Those were:

- A user seeing a list of tasks each of his or her tasks are dependent on.
- A user seeing a list of tasks dependent on each of his or her tasks.
- A supervisory role allowed to grant permissions, or assign tasks to users.

Those features are all very closely related and can be implemented almost independently from the base CAD application. Because the purpose of the implementation was to really understand and evaluate the current CAD architecture for adaptability to the multi-user requirements, the

feature that seemed the most challenging to implement was also a focus of the interface tool implementation. This challenging constraint was that of confining users to work within a predetermined geometric space.

The interface tool implementation was done in two parts, which would ideally be integrated together with each other and with many more optional features, listed as design criteria. The first part was a GUI that met the three top-rated features of the survey, an organizational tool to decompose a part into tasks and show users their list along with any dependencies. This tool also serves as an example framework for any generalized tool for organizing many users in a multi-user application. The purpose of this tool was to test the feasibility and ease-of-use for an organizational tool of this type. The second part was implemented independently and served as a basic prototype of selection filtering – the first step towards confining users to work within a predetermined geometric space. This prototype highlighted many of the requirements for multi-user architecture not currently existent in current single-user CAD applications.

4.1.1 NXCollab

The collaboration tool, NXCollab, is an independent executable, essentially a GUI that interfaces with a series of interrelated tables in a database. An administrator can use the tool to create items to store in the various tables while the other users use the GUI to query the database for the information stored in the tables. The types of information stored include tasks, users, groups, and restrictions and any details associated with any of those items. Ideally this would be integrated into the multi-user application such that when the application was started, the tool would automatically run and, with added functionality, be fully integrated into the application.

When the executable is run, NXCollab first displays a login prompt so the database can store who is currently logged in on that computer. The option to check New Task Manager (see Figure 4-1) allows a user to start a new concurrent session if checked or join the current session if not checked. The database doesn't currently have the ability to store multiple sessions at once, so you cannot choose which session to join. Ideally, you would be able to manage many sessions within the same database with a name of the session or part that you want to join. The database would keep track of users assigned to each session or part by an administrator of that session and allow the login if the credentials were accepted. The ID field was added to allow NXCollab to interface with the NXConnect database (Ryskamp, et al. 2010). The number put there should correspond with a part ID from that database and is used to populate a list of planes for that part. It has to be looked up manually.



Figure 4-1: Login Prompt

If a new session is started then the user that started the session is automatically added to the user table and granted administrative privileges. The Import Tables window (see Figure 4-2) appears allowing the user to keep some, all or none of the information from the previous session.



Figure 4-2: Import Tables Window

When starting a new session or joining a session, if the user logging in has administrative rights, the Create Task Manager Window appears (see Figure 4-3). This is the interface from which an administrator can create or edit users, groups, restrictions, securities, tasks and dependencies. It also displays a list of all the tasks that have been created along with their status, priority, who the task was assigned to, and what other tasks it is dependent on. The edit buttons are only available once something of that type has been created. The order button allows for dependencies to be added later once at least two tasks have been created. The additions can be added in any order, but creating all the restrictions first, then securities, then users, then groups, then tasks, allow for the most efficient process, because that is the order of complexity and inter-related information. Of course, if a new restriction is added later, a user can be edited to have that restriction applied to him at any time. Similarly editing of anything can be done in any order.

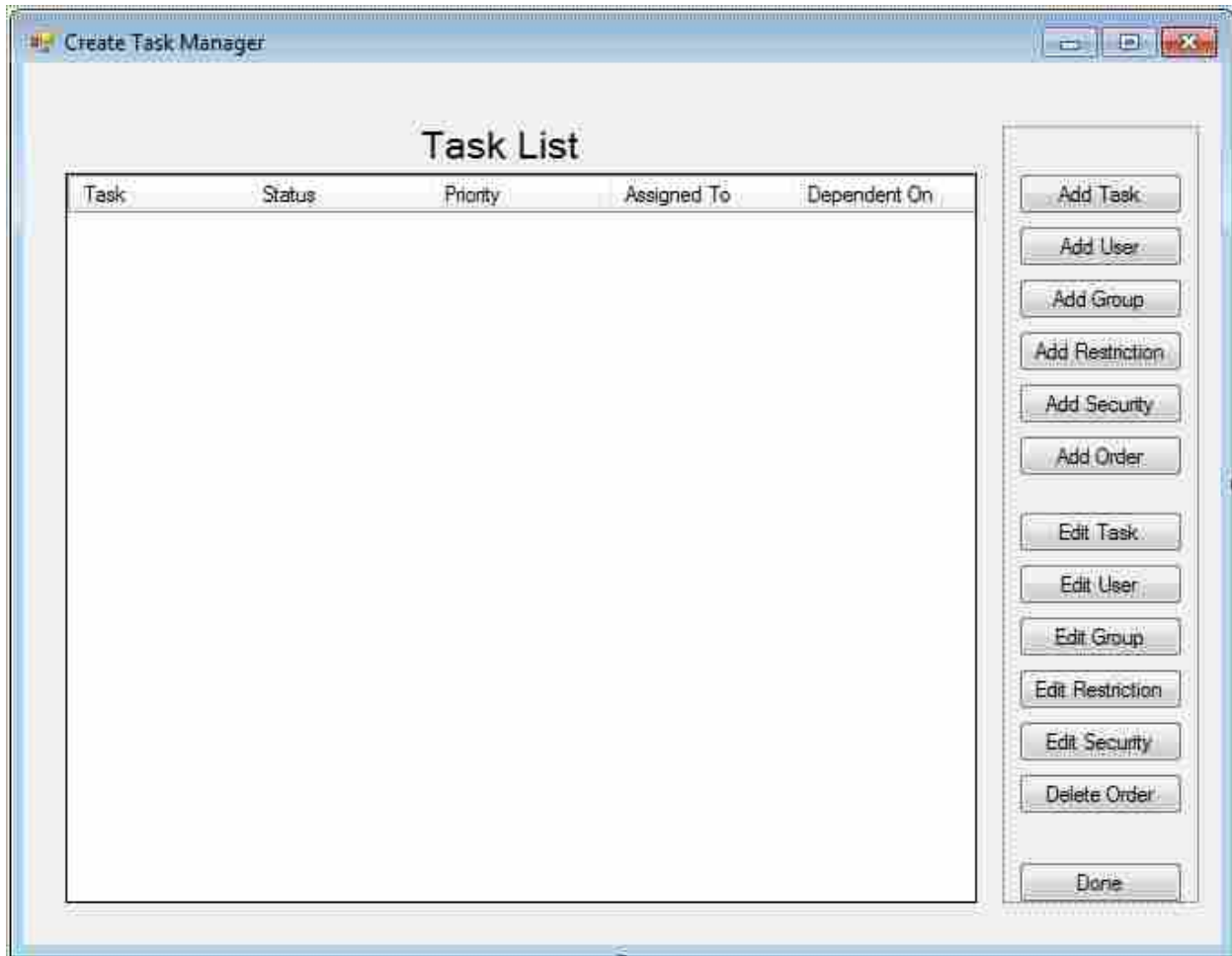


Figure 4-3: Create Task Manager Window

The Add Restriction button brings up a window (see Figure 4-4) that allows an administrator to create any kind of restriction. This may be a tool restriction, a space restriction, a time restriction, a viewing restriction, anything that has been implemented as part of NXCollab. For this implementation, it is simply a restriction name and a description of what the restriction would be, without applying any functional restrictions.

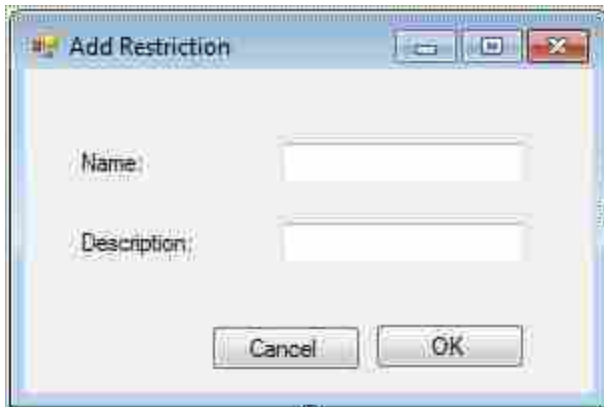


Figure 4-4: Add Restriction Window

The Add Security button brings up a window (see Figure 4-5) that allows an administrator to create security ranks that can later be assigned to users or groups. Any restrictions that were made will appear in the tool restrictions drop-down menu and allow a specific restriction to be applied to any user or group with which this security rank is associated. The security rank when applied to a task, limits who can be assigned to work on the task to anyone with a security rank equal to or greater than that of the task.

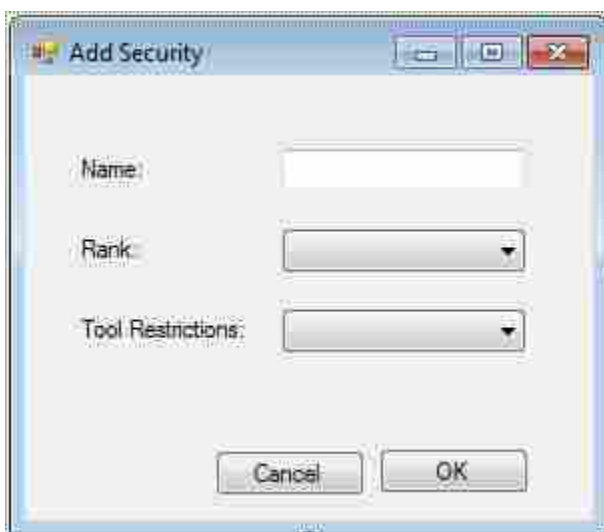


Figure 4-5: Add Security Window

The Add User Window (see Figure 4-6) allows an administrator to store a name and password for each user as well as any skills that would be pertinent to list for later trying to decide who should be assigned to work on each task. If groups have been made the user can be added to a group using this window or the user can be made an administrator. A security rank and any restrictions can also be applied to the user as part of the creation.

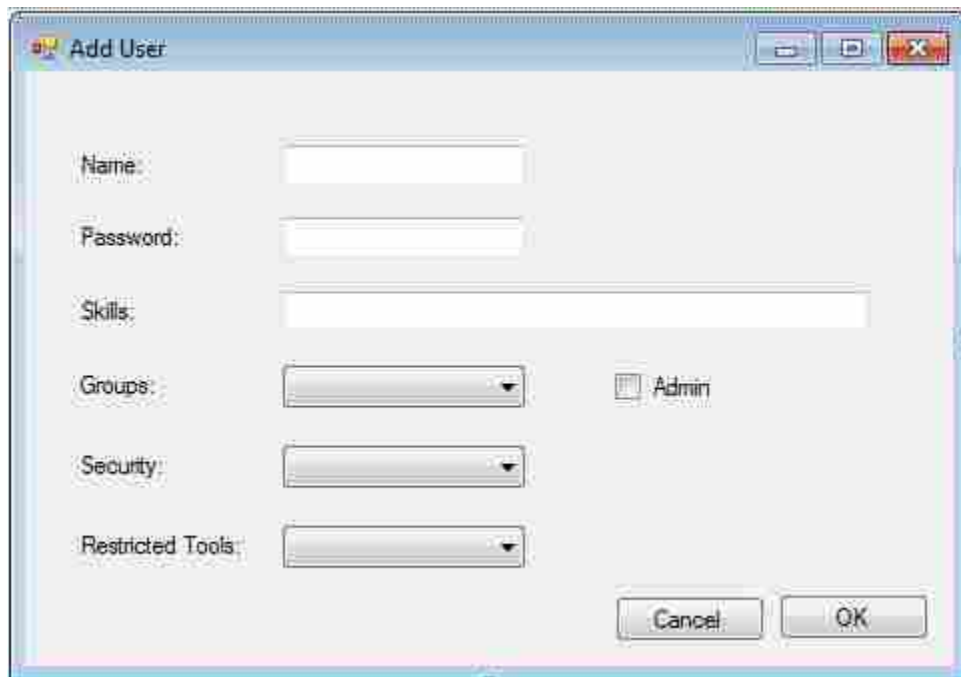


Figure 4-6: Add User Window

The Add Group window (see Figure 4-7) is a little less involved. It simply allows for a name of the group, selection of the users that should be placed in the group, and restrictions that should be applied to all members of the group to be stored.



Figure 4-7: Add Group Window

Once restrictions, securities, users and groups have been defined, the administrator can start creating and assigning tasks to users or groups. The Add Task Window (see Figure 4-8) allows for the storage of a task name and description, a placement, dependencies, a priority, a security level, an assignment, a visibility restriction and notes associated with the task. The Placement menu, when fully implemented, would automatically populate with planes or faces already created in the part, and would also have an option in the menu to create a new plane. This would be the plane to which the user would be taken when selecting the task to work on.

The Dependent On menu automatically populates with any tasks created previously. The priority options are hard-coded to have 5 values: Very Low, Low, Medium, High, and Very High. The Security menu allows a Security level to be assigned to the task. Once this level is selected, only users or groups with that security rank or higher will appear in the Assigned To menu. The Visibility menu allows for the selection of what groups are allowed to see the features of this task as they are being created. Finally, the Notes box, is just a place to store any pertinent information related to the task such as key parameters or important things to keep in mind while

working on the task. When tasks have been created, they will appear in the Task List of the Create Task Manager window.

The image shows a standard Windows-style dialog box titled "Add Task". It features a title bar with minimize, maximize, and close buttons. The main area contains the following fields from top to bottom: a text input for "Name", a larger text input for "Description", a dropdown menu for "Placement", a dropdown menu for "Dependent On", a dropdown menu for "Priority", a dropdown menu for "Security", a dropdown menu for "Assigned To", a dropdown menu for "Visibility", a text input for "Notes", and a dropdown menu for "Status". At the bottom right corner, there are two buttons: "Cancel" and "OK".

Figure 4-8: Add Task Window

The Add Order button becomes available when two or more tasks have been created. The window that appears when clicking this button (see Figure 4-9) simply allows for one task to be defined as dependent on another task. This can be done when a task is first created if they were created in the correct order.

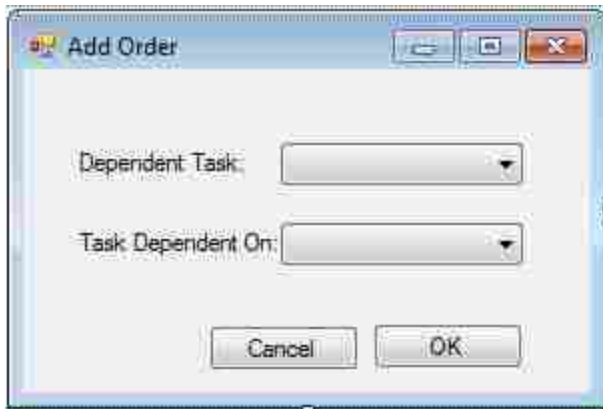


Figure 4-9: Add Order Window

Once an item has been created, editing it is simple. Clicking one of the edit buttons will bring up a window with a list of all items of that type (see Figure 4-10). The administrator can then select what item he wants to edit and click the edit button. The Edit windows for each item look the same as the creation windows except that they are already filled in with the information that was stored for them on creation. Any of the information can be edited using this menu.

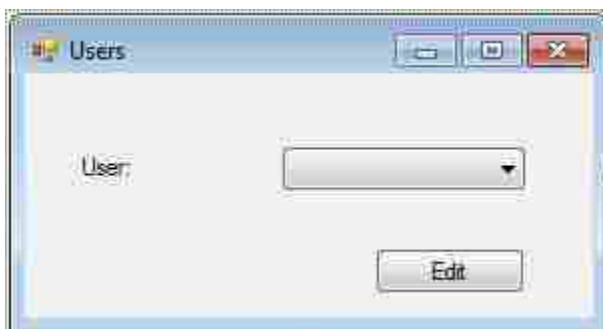


Figure 4-10: Users Window for Editing

When the administrator is done adding and creating items or reviewing the task list and the Done button is selected or when a non-administrator logs in, the User Task List (see Figure 4-11) appears. If the user is an administrator the Back to Manager button is shown in the bottom left of

the window. The Add Task button in the top left of the User Task List window brings up the same Add Task window described earlier with no restrictions if the user is an administrator. If the user is not an administrator the Add Task window still appears but the fields to assign, give a priority, apply a security rank, or apply visibility restrictions to the task are not available. The User Task List allows the user to see all of the tasks that are currently assigned to him along with the status and priority assigned to the task.

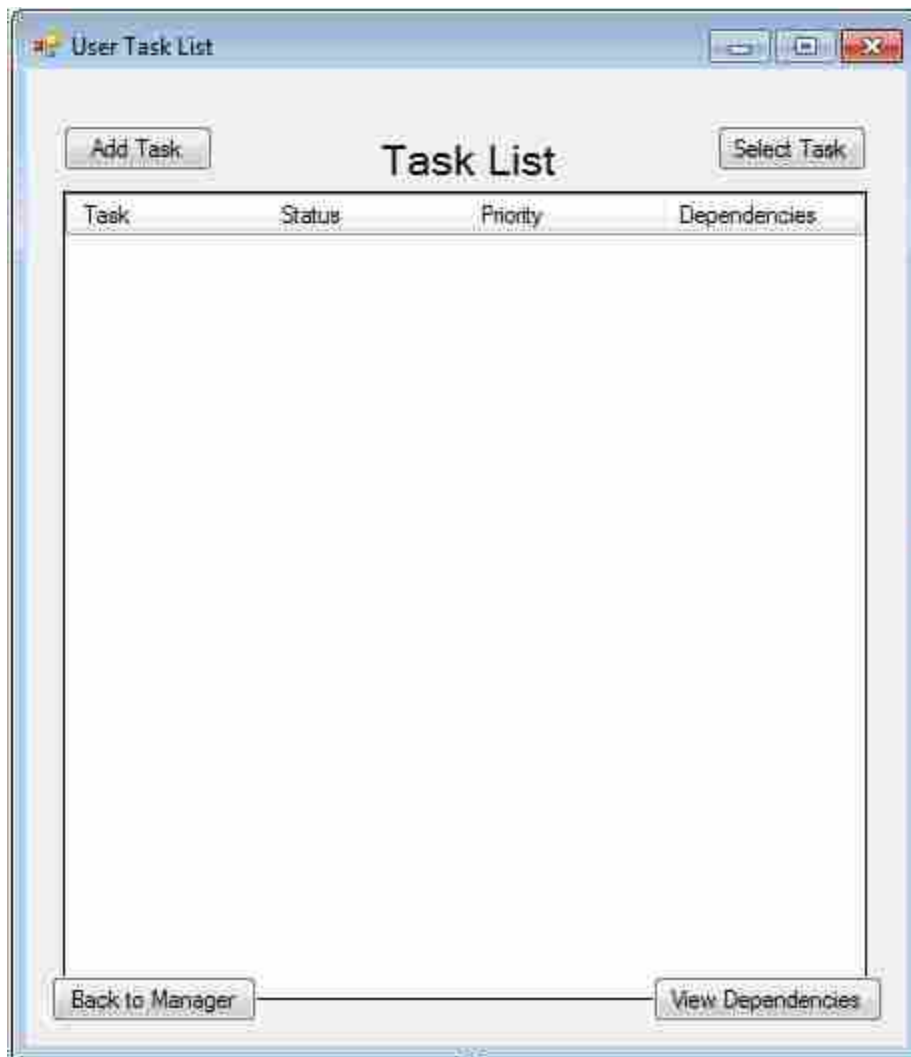


Figure 4-11: User Task List Window

A summary of all the task dependencies also appears in the table. Highlighting a task in the list and clicking on the View Dependencies button will display the two lists of dependencies (see Figure 4-12): those tasks upon which the task selected is dependent and those tasks that are dependent on the task selected.

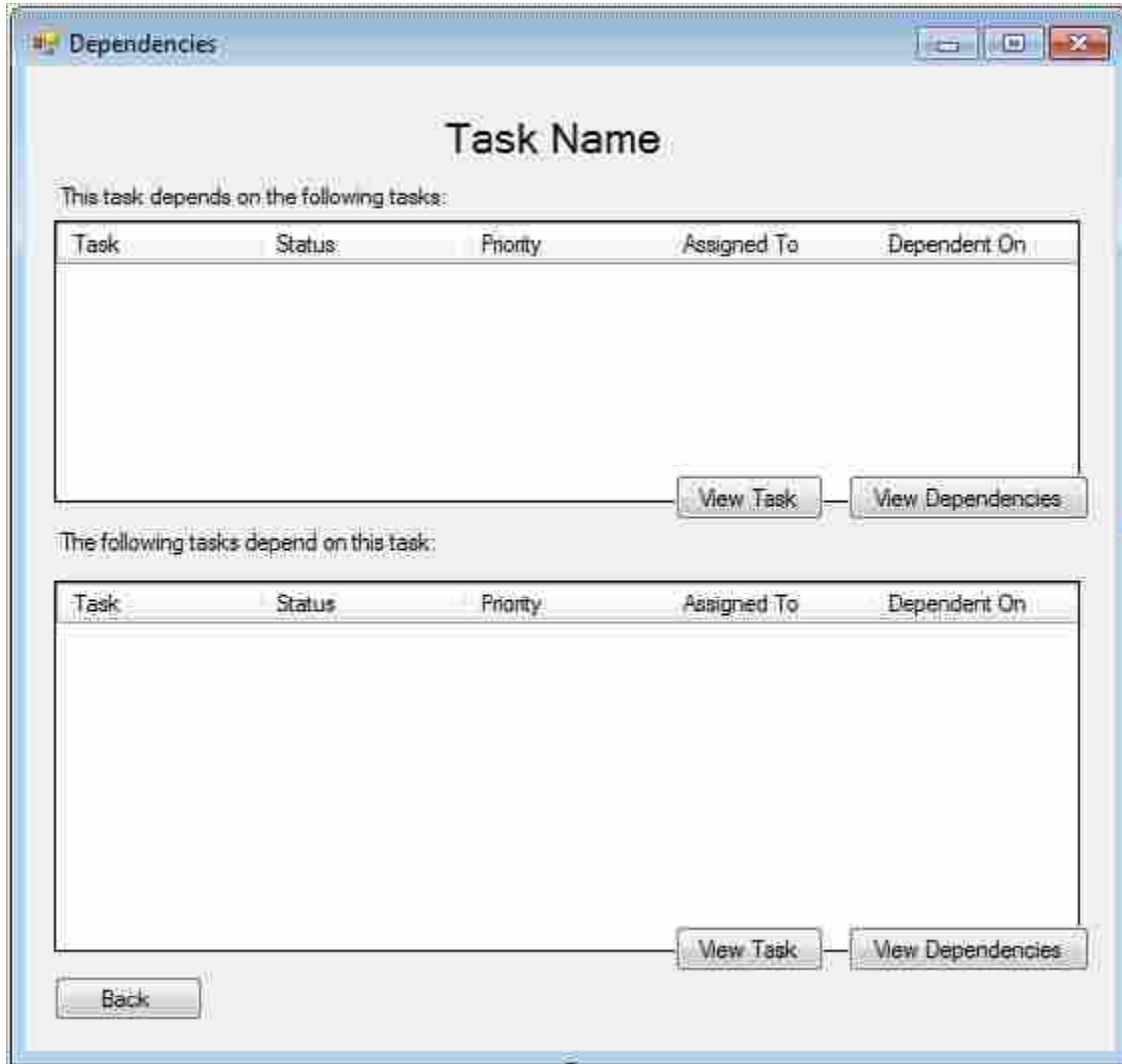
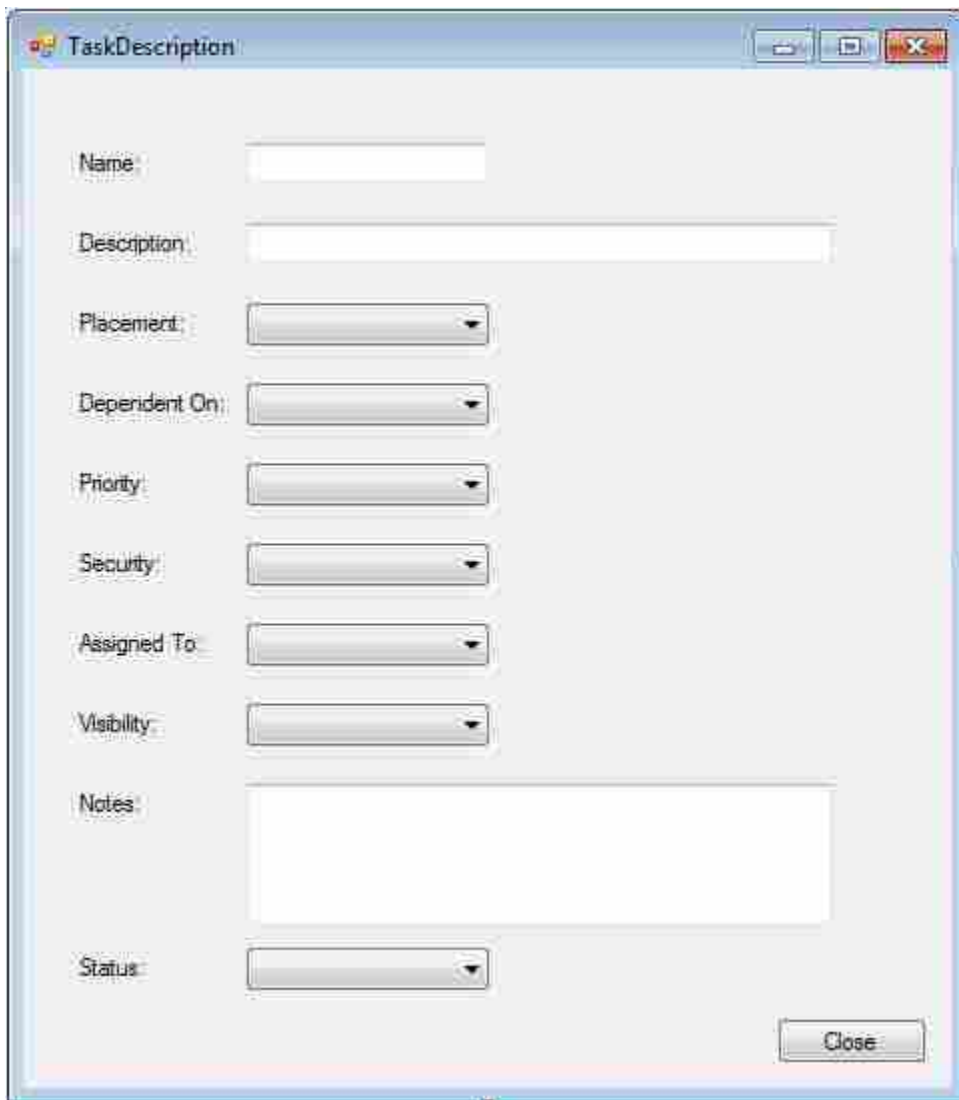


Figure 4-12: Dependencies Window

These lists show the task name, status, priority and current assignment for each task. It also shows further dependencies for each of the listed tasks. Highlighting an item in the list and selecting the View Task button will show the further details of the task. Selecting View Dependencies will bring up another dependencies window with the selected task as the focus. The Dependencies window can be closed when the user is done viewing the dependencies by clicking the Back button. Figure 4-13 shows the Task Description window.



The image shows a software window titled "TaskDescription". The window contains several input fields and a "Close" button. The fields are arranged vertically and include:

- Name: A single-line text input field.
- Description: A multi-line text input field.
- Placement: A dropdown menu.
- Dependent On: A dropdown menu.
- Priority: A dropdown menu.
- Security: A dropdown menu.
- Assigned To: A dropdown menu.
- Visibility: A dropdown menu.
- Notes: A multi-line text input field.
- Status: A dropdown menu.

A "Close" button is located in the bottom right corner of the window.

Figure 4-13: Task Description Window

From the User Task List, a user can highlight the task they want to work on and click Select Task to pull up this description. If fully integrated with the CAD system, it would also take the user to the area assigned as the Placement for the task. It would also apply any functional restrictions associated with the task to the user. The Task Description window simply shows all of the information associated with the task without allowing for any changes to the information except for updating the status of the task.

4.1.2 Selection Filtering Tool

The selection filtering tool is a GUI that runs as a .dll inside of the CAD application. The GUI remains open while the .dll is running and filters the allowable selection based on the user selected. The .dll has to be triggered manually and the filtering only lasts as long as the GUI remains open. The user selection is also manual and the constraint boundaries are hard-coded in association with each user. Ideally this program would be integrated with NXCollab as well as the application. It would also apply a more general filtering that would apply constantly and would be automatically determined based on the user credentials provided at login. Integration with NXCollab would also allow for filtering based on assigned tasks as well as universal user-associated constraint boundaries.

The selection filtering portion of the implementation is integrated with the CAD system and has a single dialog window that allows for the selection of one of 4 users. Depending on which user is selected, a selection filter is applied to all possible selections based on 4 different restrictions (see Figure 4-14). This example allows for the selection of edges and faces, which can each be considered a set, P , of points, p , where,

$$p \in P \quad (4-1)$$

And p is made up of components x , y , and z . Let X , Y and Z be the set of x -, y - and z -values of the points in the set, P such that,

$$X \subset P \quad (4-2)$$

$$Y \subset P \quad (4-3)$$

$$Z \subset P \quad (4-4)$$

where,

$$x \in X \quad (4-5)$$

$$y \in Y \quad (4-6)$$

$$z \in Z \quad (4-7)$$

The first user is allowed only to select edges and faces for which any x -value is greater than 2.15 inches.

$$\text{IF any } x \in P > 2.15, \text{ ACCEPT } P \quad (4-8)$$

The second user is allowed only to select those items for which any Z value is greater than 1.013 inches.

$$\text{IF any } y \in P > 1.013, \text{ ACCEPT } P \quad (4-9)$$

The third user is allowed only to select those items for which any Y value is less than 0 inches.

$$\text{IF any } z \in P < 0, \text{ ACCEPT } P \quad (4-10)$$

The fourth user is allowed to select only those items for which any point has an X value of less than 2.15 inches and a Y value of greater than 0 inches and a Z value of less than 1.013 inches.

$$\text{IF any } x \in P < 2.15$$

AND any $y \in P > 0$

AND any $z \in P < 1.013$, ACCEPT P (4-11)

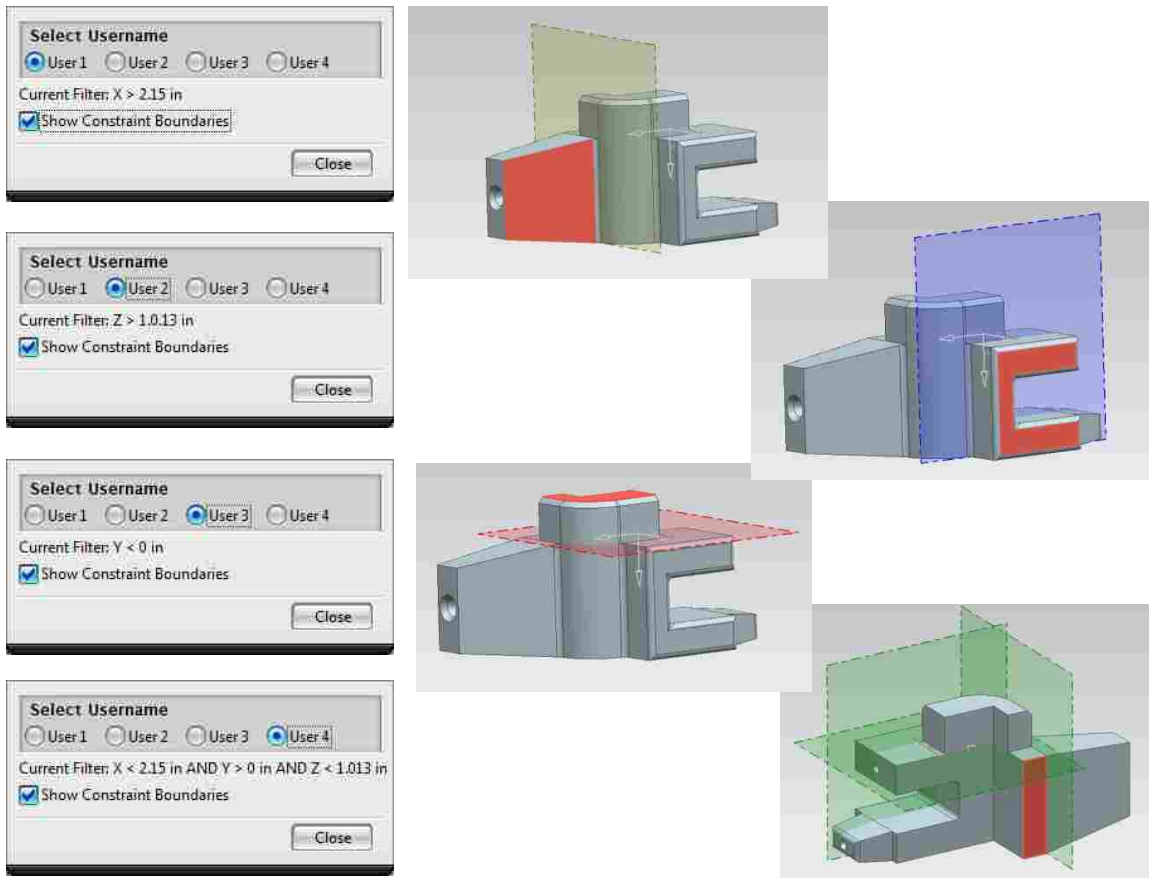


Figure 4-14: Selection Filtering Implementation

Normally a feature would highlight as the mouse hovers over it to show what would be selected if the user were to click at that moment. However, if a feature is not allowed to be selected based on the current filter applied, the features will not highlight at all when the mouse hovers over it. There is also an option in the menu to toggle on or off the visible constraint boundaries. The constraint boundaries are planes placed at the edge of the user's selection area and are colored differently for each user. When the dialog is closed, whatever is selected

becomes unselected once again and is lost. Nothing can be done with the selection while the dialog is open aside from unselecting that which is selected.

4.1.3 Benefits

There are many benefits to using NXCollab to organize users. Preliminary testing showed that it is just as fast if not faster than doing the same organization on paper. Users who tried using the tool also had no problems figuring out how to use the tool and appreciated having their task list easily distinguished from the tasks of others while still being able to see the necessary information about others' tasks. They also liked having it on the screen and easily navigable while working in the multiuser environment.

One benefit of the structure of the tool is that definitions can be as specific or as vague as the creator decides. Restrictions, Securities and Groups don't have to be used at all. Users don't have to have skills defined. Tasks don't have to have placements or priorities defined. But all of these options are included for those who would find them useful. Those who would find them inhibiting and a waste of time can skip them all together and just create user names with passwords, and vague descriptions of tasks with assignments. Adding the dependencies is a quick process that gives the users a lot of capability to understand where their tasks fit into the whole and to keep them motivated to finish those tasks others are waiting on.

The major benefit of the selection filtering tool is that it shows the potential of these multi-user systems to control very specifically and limit their interactions completely if necessary to prevent conflicts in an otherwise chaotic environment. The filtering code could also be based on any mathematical equation that could be coded. This means that cylindrical surfaces or planes at any angle could be used as constraint surfaces. There is also much flexibility in the definition of the constraint boundary. For example, a filter may compare a feature made up of a

number of points to a planar surface, and apply acceptance/rejection criteria in several ways based on that same boundary:

- Accept if all points are greater than boundary
- Accept if all points are greater than or equal to boundary
- Accept if majority of points are greater than boundary
- Accept if majority of points are greater than or equal to boundary
- Accept if any point is greater than boundary
- Accept if any point is greater than or equal to boundary
- Accept if the average of the points is greater than boundary
- Accept if the average of the points is greater than or equal to boundary
- Accept if at least one point is less than boundary and at least one point is greater than boundary
- Accept if the at least one point is less than or equal to boundary and at least one point is greater than or equal to boundary
- Reject based on any of the same criteria
- Etc.

The boundaries could also be placed anywhere, related to or unrelated to features of the part. Thus the boundaries could be set up before a part has been started or they could be applied on certain faces of a part in progress. Either way, it would be good to set them up parametrically so a boundary that is used for many users can be adjusted once and not redefined for each user.

4.1.4 Limitations

Each of the implementations has some deliberate limitations based on the infeasibility of fully implementing all possible features. Two obvious limitations are that the two are not integrated together and that NXCollab is not integrated with the CAD system directly. This means that the Placement feature is not fully implemented, meaning that although the box can automatically populate with the current planes of the part (using the NXConnect database), it does not automatically take the user to that plane when the task is select, it just tells them what plane the task is associated with. It also does not currently list faces or allow a user to create a new plane when defining the placement for the task.

Other items that are currently just informational included the restrictions and the visibility options. Ideally, the selection filtering tool would fit right in with defining restrictions and applying them to users or tasks. The visibility menu is just a place holder and doesn't actually limit the visibility of any features. For this to be implemented the architecture of the CAD system would have to be restructured to automatically associate features with the task being worked on at time of creation such that a user not assigned visibility rights for a certain task would have features associated with that task automatically hidden on their screens.

A couple of other limitations of NXCollab that could be overcome relatively easily are that the database doesn't currently have the ability to store multiple sessions at once and that the drop-down menus for all items are single selection menus. The database should be developed to store a name or id for a session and allow users to join a particular one without confusing all of the stored data for each session. Also, because all the drop-down menus are single selection, multiple restrictions cannot be applied to a user, group or task; a user cannot be assigned to multiple groups; and multiple dependencies cannot be applied to a single task. Fortunately, the

database is structured such that it could support each of those functions. Unfortunately, the interface was not designed and implemented to allow for it.

A big limitation of the selection filtering tool is that the constraint boundaries are hard coded as an example. Ideally it would be implemented into NXCollab and the constraint boundaries would be defined by the administrator as optional restrictions and would be applied to specific users, groups or tasks.

Besides the deliberate limitations, the implementation particularly of the selection filtering tool brought to light some of the inherent limitations of the current CAD architecture. The function used to filter the selection requires an input argument called a selection handle. This selection handle is automatically created by each new dialog that is open; there is no way to create a selection handle without it being associated with a particular dialog box. This means that the selection filter can only work while the dialog box is open and the selection itself unloads at the closing of the dialog box. A custom dialog box was made to showcase the selection filtering, but doing anything with the selection is impossible unless the function is integrated into the code and GUI of the dialog box. Thus, to implement a useful selection filter, the code would have to be written in to each dialog box the CAD system used while selected things, and that still wouldn't cover the situations where features are selected outside of having a dialog box open. Filtering the access to tools or buttons is not covered by this research (see Chapter 5) but would likely have similar difficulties in the current architectures.

Another serious limitation of the current CAD architecture is the difficulty of accessing geometric data for each feature. Simple features like points and lines are easily compared to a constraint boundary, but edges and faces are associated with the features they are made from rather than their makeup geometry. Thus, to find points to test against the constraint boundaries,

a bounding box function must be used. This function essentially returns 6 values, the max and min values of the feature for the x, y, and z directions. Thus all you have to compare against your boundary is two points that in many cases aren't even on the surface to which you are comparing the constraint. This means that the comparisons end up not being very accurate unless the features are made with edges aligned with the coordinate axes.

Beyond the selection filtering difficulties, it has not yet been determined how to completely confine a user to a geometric volume. For example, how would the system prevent a user from creating a feature outside of the boundary constraints? Another difficulty is representing the boundaries accurately and without distracting from the modeling environment. Because there are so many ways to define the boundaries, there has to be equally as many ways to represent what kind of boundary is being applied to a user in an easily-, quickly-, and universally-understood and non-distracting way. These are both topics recommended for further work in Chapter 5.

4.2 Tools Used

The development of NXCollab was done in C# using Microsoft Visual Studio 2008 Version 9.0.30729.1 SP with Microsoft .NET Framework Version 3.5 SP1. There was also some limited interaction with the tables of the NXConnectDB for the NXConnect prototype made previously. The NXConnectDB database has since been restructured dramatically and that interaction no longer works.

The development of the selection filtering tool was also done in C# using Microsoft Visual Studio 2008 Version 9.0.30729.1 SP with Microsoft .NET Framework Version 3.5 SP1.

The code was built for Siemens NX 6.0.4.3 using their NX Open for .NET API. Initial code with the framework for using the selection filter was provided by Siemens programmers.

All development and testing of the programs was done on Windows 7 Enterprise with Service Pack 1.

5 CONCLUSIONS AND RECOMMENDATIONS

To ensure that multiple users in a single environment operate collaboratively rather than chaotically, using an interface tool such as NXCollab is an effective method. The organization it provides along with the flexibility to adapt to varied levels of use make it a suitable choice for any company that uses CAD. Beyond application to the CAD environment, the method used to create NXCollab is a valid method for any multi-user environment. An organizational tool with various constraining features optionally integrated into it can logically decompose a space in any multi-user virtual environment.

Beyond the collaboration tool, there are many other areas of research related to this that still need extensive development, both in the general decomposition field and in the multi-user CAx Applications arena. Visual representation of boundaries, geometry creation constraints, and tool constraints are CAx specific research topics, where general decomposition methods and methods for direct conflict resolution are huge areas that would affect all multi-user virtual environments.

5.1 Conclusions

This research meets all five objectives stated in the introduction and shows that there is a feasible, flexible, scalable and relatively simple method to constrain users to work collaboratively:

1. A generalized method for model decomposition of single part files by developing administrative controls for parametrically dividing a model into tasks and user assigned regions was defined based on the research and survey results and discussed in Chapter 3.
2. Specific modeling constraints for multi-user CAx applications were outlined. The Selection Filtering tool demonstrated geometric constraints while NXCollab showed feature-based constraints by breaking the model into features using tasks. The method also introduced the ideas of functional constraints by adding tool and other restrictions to a user or task and order-specific constraints by allowing for dependencies between tasks.
3. The NXCollab method allows an administrator to assign users to specified tasks and regions and the generalized method discusses limiting their access and interactions with other regions of the model by implementing various constraints.
4. The method for maintaining model continuity between user regions is inherent to the feature-based decomposition because the features still interact with each other in the same manner that they would in a single-user environment. When users are confined to geometric volumes, the shared boundaries have to be defined through coordination and communication between users. The notes attached to tasks could hold key information regarding shared boundary information. The administrator could also task a single user with defining boundary regions.
5. NXCollab demonstrated the effectiveness of the decomposition method in coordinating multiple users while the Selection Filtering tool demonstrated the

capability of implementing more advanced constraints and outlined current architectural limitations.

The most difficult aspect of this research lies with specific constraint features that may require architectural changes to the underlying application. This is certainly the case for geometric constraints within current CAD architectures.

5.1.1 Architectural Changes

In general it is probable that whatever application might be evolving from a single-user to a multi-user scope will have to undergo some architectural changes to encompass any advanced constraint features. For the case of CAx architectures, some important constraint abilities are unrealistic with the current architecture design. The architecture needs to allow for a more general control over the selection and viewing of features with the capability of filtering based on any criteria. It would be best if that architecture were to allow for easy and direct access to all of the base geometry of every type of selectable feature. It may even mean that features are not stored in a hierarchical and chronological tree structure, but in a more geometrically-defined fashion.

Other changes to the architecture may be due to visualization in the multi-user environment. Because collaboration requires that users be aware of their surroundings, it is important to find the best way to show a user what is going on in the vicinity around him without the possibly distracting movements of multiple cursors all over the screen.

5.1.2 Tool Integration

An important aspect of a multi-user application constraint tool is that it is integrated with the application itself. A general constraint tool as an add-on to the application would not be able to provide the customized types of constraints that are required for collaboration to be most effective. The tool should be made custom to and in tandem with the multi-user application itself, so the application architecture is designed with the required capability to support the desired constraint features and so that the tool does not stick out as distracting or clumsy.

It is important that the tool not only work seamlessly within the application, but that it blends in with the application theme as part of the complete package. This will take the focus off of the tool so that it doesn't seem like extra busy work required to get to the modeling, but as a useful tool to help with more efficient modeling.

5.1.3 Benefits

The biggest benefits of using an organizational tool as the primary constraint mechanism is that it has the capability of being as restrictive as necessary or as unrestrictive as necessary. Many companies are distributed throughout the world and have a varied workforce for which language and customary barriers can cause major communication problems. Having hard constraints such as boundary volumes within which a user is confined, could help prevent major conflicts due to miscommunication. Other companies often have highly communicative teams that just need to work together quickly to finish up a design and the extra constraints would be too prohibitive to be helpful, where moving more fluidly throughout the part would work better.

Another advantage to the organizational tool is that it requires at least some forethought and acts as built-in documentation. It helps an administrator to break up the modeling into tasks

and to document the intentions and assignments clearly in a manner integrated with the design environment. Users can then easily find what they are supposed to be working on and what other work is dependent on theirs, increasing motivation to complete tasks.

5.1.4 Drawbacks

The most obvious drawback of an organizational constraint tool is that the process of breaking a model into the required tasks is difficult and not-well defined. Until further research makes that aspect of the decomposition more systematic, many product development groups may not be ready to embrace the full capability of a tool like this. Fortunately, they can define things as vaguely as necessary to get started, but the tool will only be as helpful as the level of definition used.

Other drawbacks are that many of the most advanced constraint options will require major architectural changes to the single-user application for full implementation. In the case of constraint boundaries, it may not be helpful to allow any surface to be a constraint boundary. The higher-order the surface is, the more difficult it will be to compare the geometry to. Separating users by physical boundaries might also disrupt the continuity of the regions. It may require that the users on either side communicate well to maintain the continuity between regions, or it may require a third user to be responsible for working on the boundary regions.

5.2 Recommendations for Further Work

The concept of geometrical boundaries is a complex one that requires much more research. Other types of constraints will require further work as well. Another huge area of research is theoretical complex part decomposition. Finally, it is important that beyond

preventative conflict resolution, that direct conflict resolution for a multi-user virtual environment be addressed in great detail.

The thesis demonstrates the usefulness of the method in the CAD environment and could easily be applied to other engineering environments such as analysis or manufacturing programs. A study should be conducted looking at Product Lifecycle Management (PLM) systems and optimization frameworks as well.

5.2.1 Visual Representation of Boundaries

This and the research of Xu needs to be extended to include the representation of user-boundary regions (Xu 2010). Whenever geometrical boundaries are used, it is pertinent that the visualization of those boundaries is clear and not inhibitive. As with all of the constraint features, there has to be option to allow for the variety of users that will need to operate in the system, some may not want to have the constraint boundaries of others visible, others may not even want to see their own constraint boundaries, especially if there are many or complex boundaries that may obscure the view of features they would prefer to see.

Once the on/off viewing options are sufficiently defined, the more difficult problem is that of displaying the boundaries in a way that demonstrates what type of boundary it is. There are many different ways for a boundary to be defined. When dealing with three dimensional objects compared to surfaces, the constraint can require all parts of the object are on one side of the boundary or the other. It can include or exclude objects that touch the boundary or cross the boundary. It can require that objects touch or cross the boundary or remain within a tolerance of the boundary.

Finally, besides the type of boundary a surface represents, it must also be clear what side of the boundary, unless it is the boundary area itself, the user is allowed to work on. Obviously,

if the selection filter were properly functioning and there was geometry on either side of the boundary, it would be easy to figure this out by hovering over the different features, but if nothing had yet been created, the boundaries would have to be more telling.

5.2.2 Creation Boundaries

When nothing has been created, or when there is still substantial creation to be done, it would, in many cases, still be important to confine a user to work within a certain geometric volume. Limiting the placements of objects as they are being created is an area that hasn't been investigated at all. It presents a much more complicated problem than just filtering the selection, which in itself proved to require extensive changes. Creation boundaries would likely be very different in structure for different applications.

5.2.3 Tool Constraints

Although the idea of tool constraints was presented as an important constraint feature from the beginning of this research, the actual investigation of how this would be implemented has yet to take place. The idea behind this is that as part of the constraint tool, users, groups or tasks could have tool restrictions associated with them. A certain user or group would then only be allowed to use certain tools already inherent to the application or while working on a specific task only certain tools would be available for use.

5.2.4 Complex Part Decomposition

A hugely important area of research is that of complex part decomposition from the theoretical standpoint. This process is the precursor to using the constraint tool and is largely undefined. It is much easier to decompose a part that has already been created for analysis or

manufacturing, but decomposing a part before it has been designed is a much more abstract problem. Being able to take an idea and break it apart into features or tasks is required for modeling and is often done by sketches or defined while modeling. It is likely, that not having a defined approach to decomposing the part into tasks for modeling is the main contributor to excessive feedback loops and iterative processes in honing a design.

5.2.5 Direct Conflict Resolution

Finally, the issue of direct conflict resolution is possibly one of the most complex problems related to this research. Because it is the more complex subject, this research focused on preventing conflict and avoiding this difficult topic as much as possible. Unfortunately, despite the best preventative efforts, there will always be direct conflicts to deal with, and defining multiple methods for intelligently doing so will be increasingly important with more and more users working together in the same environment.

5.2.6 PLM Applications

Beyond application to a single design or analysis program, this thesis can more generally apply to PLM systems and optimization environments. For example, in a PLM system the collaboration tool would live outside of the various programs used for the project and run in parallel with them, dividing a project into tasks across multiple programs or platforms. One of the key attributes of a task would then be what program or programs the task is associated with. The task could also apply optional constraints or store other attributes based on the program with which the task is associated. Then when selecting a certain task to work on, a program may open and load a particular file or do any number of customizable operations.

The boundary and tool constraints between design regions could also more generally be applied to the PLM space. While working on certain tasks, a user could be restricted from using certain programs or opening certain files or folders. A group may not be able to access or may be allowed only viewing rights to certain documents. A program may be configured to communicate with the collaboration tool and start up in different modes based on the user credentials and clearance level. Just as in a single program application, in a PLM environment, a user may have restrictions placed on him universally granting only limited access to project information, perhaps even hiding the existence of files or folders from the user. A 3rd party viewer could then be involved without seeing any of the ensuing project development and only having access to specifically assigned documents. Further investigation is required to better understand the capabilities and limitations of applying the method described in this thesis to larger multi-application environments.

REFERENCES

- Augustina, F. Liu, S. Xia, H. Shen, C. Sun. "CoMaya: Incorporating Advanced Collaboration Capabilities into 3D Digital Media Design Tools." *CSCW'08*, November 8-12. (2008): 5-8.
- Bu, J., B. Jiang, and C. Chen. "Maintaining Semantic Consistency In Real-Time Collaborative Graphics Editing Systems." *IJCSNS international Journal of Computer Science and Network Security* 6, no. 4 (2006): 57-61.
- CADAZZ. 2004. <http://www.cadazz.com/cad-software-history.htm> (accessed January 7, 2011).
- Cera, C., I. Braude, I. Comer, T. Kim, J. Han, and W. Regli. "Hierarchical Role-Based Viewing for Secure Collaborative CAD." *DETC'03 2003 ASME Design Engineering Technical Conferences*. 2003. 2-6.
- Chan, C., and S. Tan. "Volume Decomposition of CAD Models for Rapid Prototyping Technology." *Rapid Prototyping Journal* 11, no. 4 (2005): 221-234.
- Chen, L., Z. Song, and L. Feng. "Internet-Enabled Real-Time Collaborative Assembly Modeling Via an e-Assembly System: Status and Promise." *Computer-Aided Design* 36 (2004): 835-847.
- Chong, C., A. Kumar, and K. Lee. "Automatic Solid Decomposition and Reduction for Non-Manifold Geometric Model Generation." *Computer-Aided Design* 36 (2004): 1357-1369.
- Cox, J., W. Charlesworth, and D. Anderson. "Domain Composition Methods for Associating Geometric Modeling with Finite Element Modeling." *Association for Computing Machinery*, 1991: 443-454.
- Dempski, K., B. Harvey, and B. Korytkowski. "Multi-User Affordances for Rooms With Very Large, Interactive, High-Resolution Screens." n.d.
http://www.accenture.com/NR/rdonlyres/EB097C88-C0F9-484E-890A-E5E7571A7154/0/multiuser_affordances.pdf.
- Fuh, J., and W. Li. *Advances in Collaborative CAD: The State-of-the-Art*. Singapore Institute of Manufacturing Technology, 2004.
- Gould, Lawrence S. *Multi-CAD Environments: New Software Technologies are Focusing on Editing and Sharing 3D Geometry*. September 1, 2008.

<http://www.thefreelibrary.com/Multi-CAD+environments:+new+software+technologies+are+focusing+on...-a0200560685>
(accessed January 7, 2011).

- Hoffman, C., and R. Juan. "Erep - An Editable High-Level Representation for Geometric Design." In *Geometric Modeling for Production Realization*, by P. Wilson, M. Wozny, & M. Pratt(eds), 129-164. North Holland, 1993.
- Jing, S., F. He, S. Han, X. Cai, and H. Liu. "A Method for Topological Entity Correspondence in a Replicated Collaborative CAD System." *Computers in Industry* 60, no. 7 (2009): 467-475.
- Lai, Y. "A Constraint-Based System for Product Design and Manufacturing." *Rovotics and Computer-Integrated Manufacturing* 25, no. 1 (2009): 246-258.
- Lin, K., D. Chen, C. Sun, and G. Dromey. "Maintaining Constraints in Collaborative Graphic Systems: The CoGSE Approach." *ECSCW 2005: Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work*. Paris, 2005. 185-204.
- Lu, S., and J. Jian Cai. "A Collaborative Design Process Model in the Socio-Technical Engineering Design Framework." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 15, no. 1 (2001): 3-20.
- Mix, K., C. Jensen, J. Ryskamp. "Using Global Communication Trends for Automated Design Rationale Capture." *Proceedings of TMCE 2010 Symposium*. Italy, 2010: 585-596.
- Olson, I., Z. Leong, U. Wilensky and M. Horn. "'It's just a toolbar!' Using Tangibles to Help Children Manage Conflict Around a multi-Touch Tabletop." *TEI'11*, January 22-26. (2011): 29-36.
- Ram, D., N. Vivekananda, C. Rao, and N. Mohan. "Constraint Meta-Object: A New Object Model for Distributed Collaborative Designing." *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 27, no. 2 (1997): 208-221.
- Red, E., C. Jensen, V. Holyoak, F. Marshall, and Y. Xu. *v-CAX: A Research Agenda for Collaborative Computer-Aided Applications*. Provo: Brigham Young University, 2009.
- Ryskamp, J., C. Jensen, K. Mix, and E. Red. "Leveraging Design Rationale to Improve Collaboration Co-Design CAD Environments." *Proceedings of TMCE 2010 Symposium*. Italy, 2010: 475-486.
- Sun, W., T. Ma, and Y. Huang. "Research on Method of Constraint Conversion in Feature-Based Data Exchange Between Heterogeneous CAD Systems." *Journal of Mechanical Science and Technology* 23 (2009): 246-253.

Sun, C., S. Xia, D. Sun, D. Chen, H. Shen and W. Cai. "Transparent Adaptation of Single-User Applications for Multi-User Real-Time Collaboration" *ACM Transactions on Computer-Human Interaction*, Vol. 13, No. 4 (2006): 531-582.

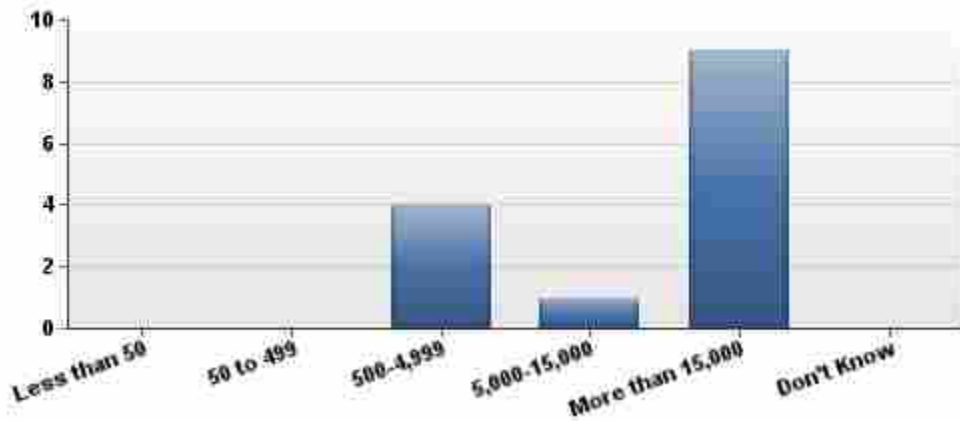
Wei, y., and P. Egbelu. "Process Alternative Generation from Product Geometric Design Data." *Institute of Industrial Engineers Transactions* 32 (2000): 71-82.

Xu, Yue. *A Flexible Context Architecture for a Multi-User GUI*. Brigham Young University, 2010.

APPENDIX A. SURVEY RESULTS

14 respondents

1. Industries surveyed: Aerospace, Automotive, Education, Manufacturing, CAx Software Vendor.
2. Company Sizes were: 500-4999 employees (29%), 5,000-15,000 (7%), more than 15,000 employees (64%)



3. 10/14 (71%) Have members of the product development workforce located outside the U.S.
4. The following corresponds to the percentage of companies that had workers in the following countries out of the 10 international companies:
 - India 60%
 - Canada 50%
 - United Kingdom of Great Britain and Northern Ireland 50%
 - Mexico 40%
 - Brazil 30%
 - China 30%
 - Czech Republic 30%
 - Germany 30%
 - Japan 30%
 - Netherlands 20%
 - Australia 20%
 - France 20%
 - Israel 20%
 - Italy 20%
 - Republic of Korea 20%

- Russian Federation 20%
- Austria 10%
- Belgium 10%
- Democratic People's Republic of Korea 10%
- Hungary 10%
- Ireland 10%
- New Zealand 10%
- Norway 10%
- Poland 10%
- Saudi Arabia 10%
- Singapore 10%
- South Africa 10%
- Spain 10%
- Sweden 10%
- Switzerland 10%

5. The following corresponds to the average percentage of workers the 10 international companies had in each country:

- United States 63.60%
- Republic of Korea 9.90%
- India 5.70%
- United Kingdom of Great Britain and Northern Ireland 4.90%
- Canada 3.80%
- Mexico 3.00%
- China 2.20%
- Germany 1.60%
- Brazil 0.60%
- Czech Republic 0.60%
- Japan 0.50%
- Israel 0.40%
- Italy 0.40%
- Australia 0.30%
- France .30%
- Russian Federation 0.30%
- Singapore 0.30%
- Netherlands 0.20%
- Poland 0.20%
- Austria 0.10%
- Belgium 0.10%
- Democratic People's Republic of Korea 0.10%
- Hungary 0.10%
- Ireland 0.10%
- New Zealand 0.10%
- Norway 0.10%
- Saudi Arabia 0.10%
- South Africa 0.10%
- Spain 0.10%
- Sweden 0.10%
- Switzerland 0.10%

6. Complex Parts examples (Examples given were A Front Frame, An Engine Block, A Blender Body):

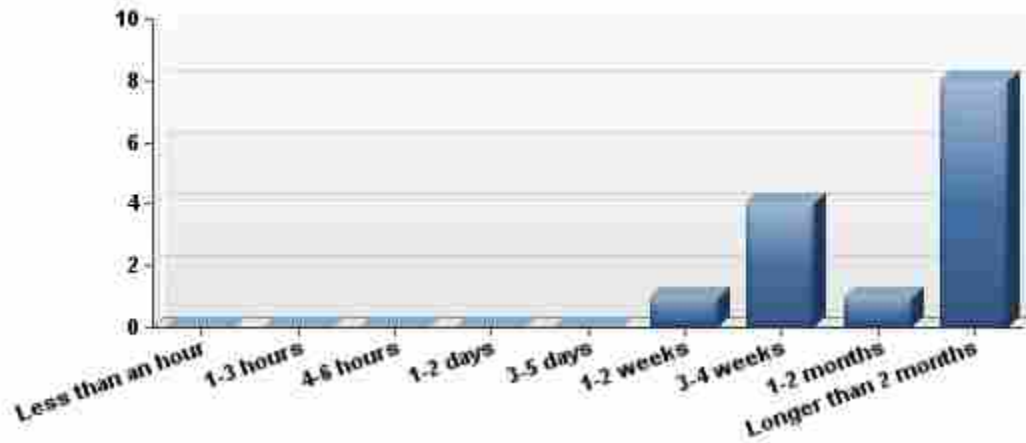
- Those listed as examples and body sheet metal - design and engineering are spread across world

- Nozzle contours
- Airplane components
- The upright component within a front suspension
- Aircraft parts
- A combustion chamber
- Gas turbine airfoils
- Turbine blade/vane
- An Engine Block
- Rocket Motor Nozzle
- Automotive Hub and Bearing Assembly
- A Front Frame, Combustors, Bladed disks, Bearing housings
- Integrally Bladed Rotor

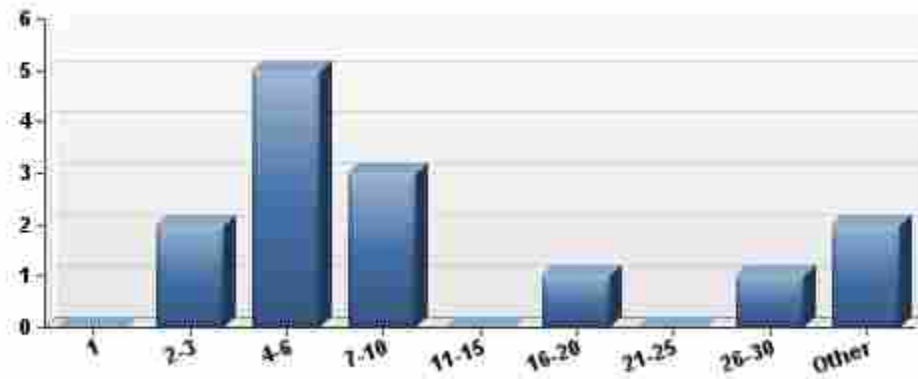
7. What about them is complex:

- Design is an aspect of appeal and aesthetics. Engineering is an aspect of durability and safety. Yet as design evolves, engineering must be able to quickly adjust packaging and determine QRD.
- Many complex facets designed and analyzed from multiple perspectives.
- Multiple users are required to: get loads, get geometry, perform part integration, build FE models, perform optimization, certify components.
- The modeling is difficult, lots of features and blends. The analysis is difficult mainly because of the meshing and loading phase of the preprocessing. The manufacturing is difficult because of all the tool paths and multiple setups required.
- It is a large monolithic structure that requires many users to create ply definition.
- Airplanes are complex. Requires lots of workers.
- The design function is really an integration function that includes disciplines like Aerodynamics, Structures, Product Definition, Validation and others. All of these different disciplines will use the geometric description to evaluate how well the design meets criteria that each discipline is managing.
- Key component in Jet engine: subject to high temperatures and stresses; component failure would be catastrophic, etc.
- Detailed geometry is dependent on multiple specialized engineering disciplines (flow, combustion, structural, thermal, dynamics, tolerancing, manufacturing, etc.) and no single engineer is an expert in all of these.
- It is complex because of the many different materials (metals, thermal-ablative insulators) needed to create a variety of parts that need to operate in a very harsh (high temperature, chemically reactive) environment.
- The assembly typically contains 10 components. The components have various types of interfaces with respect to each other including: Interference fits (requires tolerance stack study), rolling contact (requires stress and predicted fatigue life evaluations), fastening with screws (requires evaluation of thread fit and torque range), sliding contact (requires deformation, pressure, and lubrication evaluation), and clamping contact (requires stress, deformation, and strength evaluation).
- Precision manufacture is required, extensive welding and/or coatings, third and fourth decimal place tolerances are required.
- Complicated
- Multiple discipline inputs (Design, Structures, Heat Transfer, Aero). Intensive analysis (Harmonic Cyclic Symmetry modal analysis)

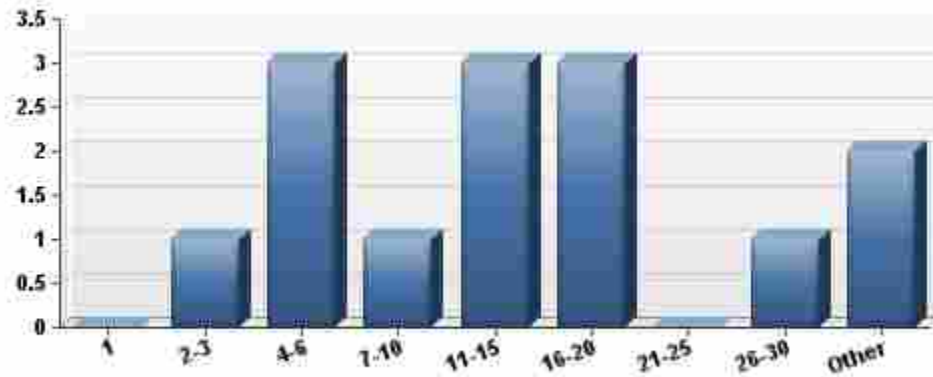
8. Part Development takes: 1-2 weeks (7%), 3-4 weeks (29%), 1-2 months (7%), Longer than 2 months (57%)



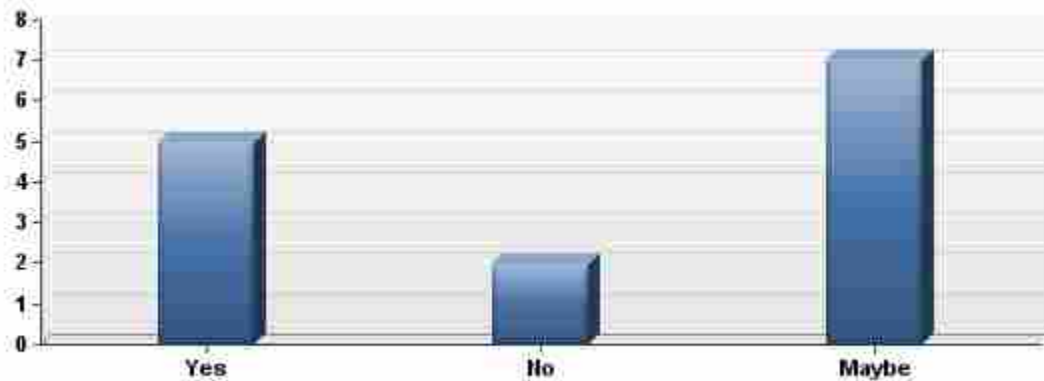
9. Average workers assigned to complex part: 2-3 (14%), 4-6 (36%), 7-10 (21%), 16-20 (7%), 26-30 (7%), 100s(14%)



10. Most workers assigned to complex part: 2-3 (7%), 4-6 (21%), 7-10 (7%), 11-15 (21%), 16-20 (21%), 26-30 (7%), 100s (14%)



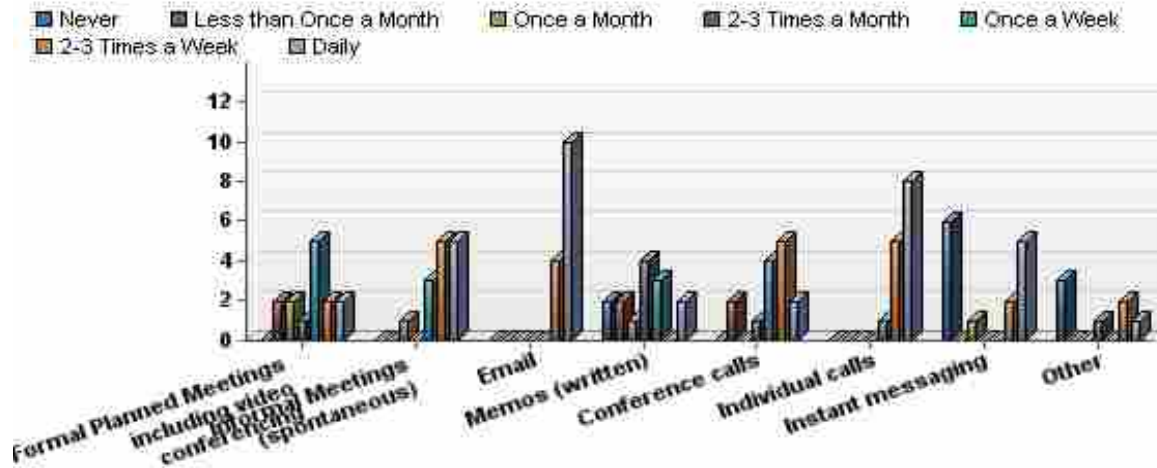
11. 36% would assign more workers to work on a part if they could, 14% would not, and 50% might.



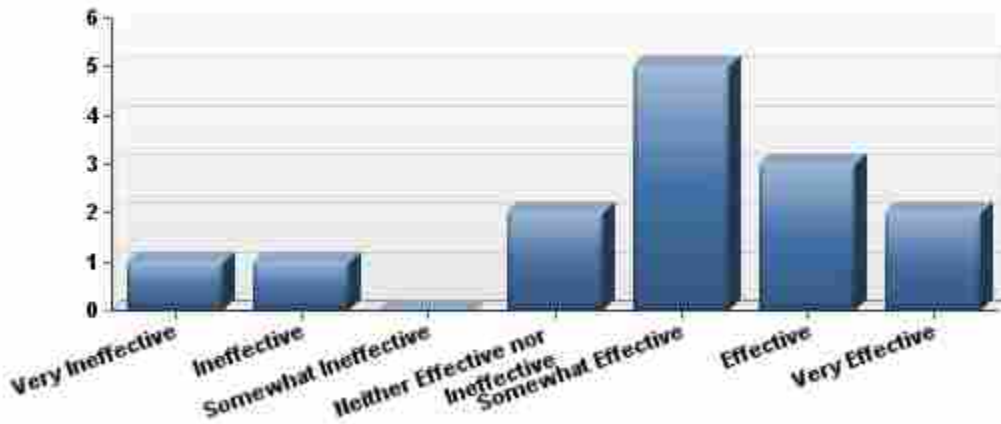
12. The following is a list of communication methods and were rated as follows: Never (1), Less than Once a Month (2), Once a Month (3), 2-3 Times a Month (4), Once a Week (5), 2-3 Times a Week (6), Daily (7)

Type	Response Range	Most Popular Response	Average
Formal Planned Meetings including video conferencing	2 to 7	5 (5 respondents)	4.64
Informal Meetings (spontaneous)	3 to 7	6 (5 respondents)	5.93
Email	5 to 7	7 (10 respondents)	6.71
Memos (written)	1 to 7	4 (4 respondents)	3.86
Conference calls	2 to 7	6 (5 respondents)	5.14

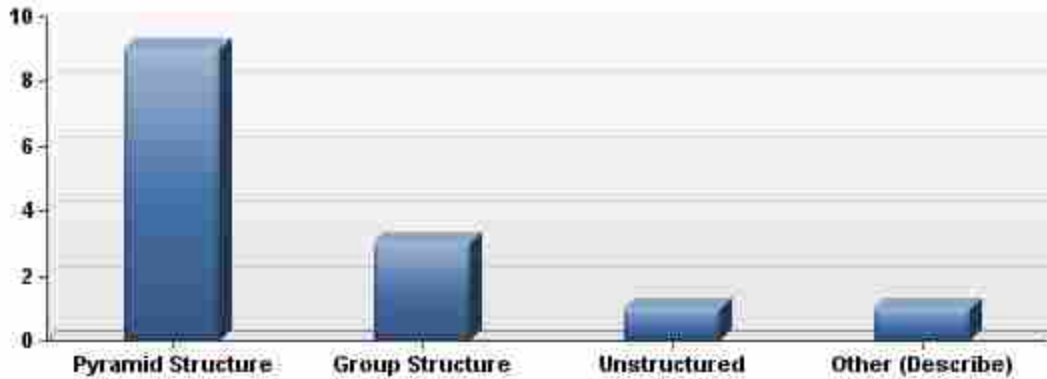
Individual calls	5 to 7	7 (8 respondents)	6.50
Instant messaging	1 to 7	1 (6 respondents, 7 (5 respondents)	4.00
Other: WebEx, net meetings	1 to 7	1 (3 respondents), 6 (2 respondents)	3.71



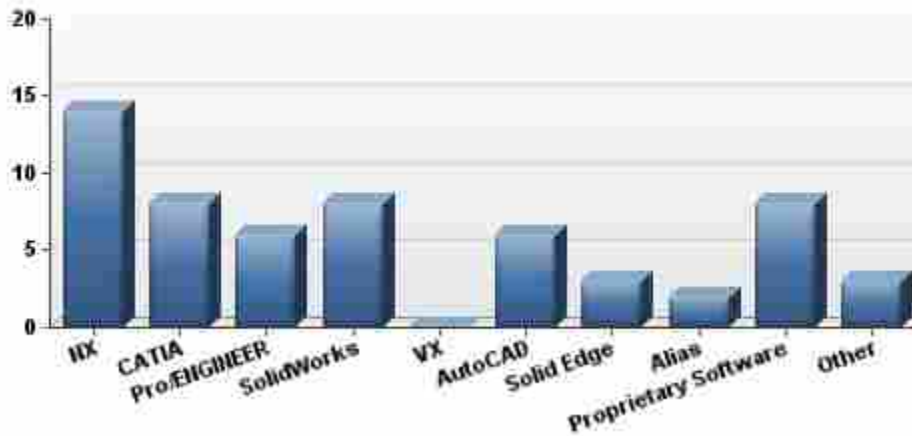
13. Communication was rated as: Very Ineffective (7%), Ineffective (7%), Neither Effective nor Ineffective (14%), Somewhat Effective (36%), Effective (21%), Very Effective (14%)



14. Project Team Management was structured as: Pyramid (64%), Group Structure (21%), Unstructured (7%), Other: It is a group structure where all of the workers have a lot of communication between each other and provide feedback to the manager. It is not unstructured as pictured. (7%)



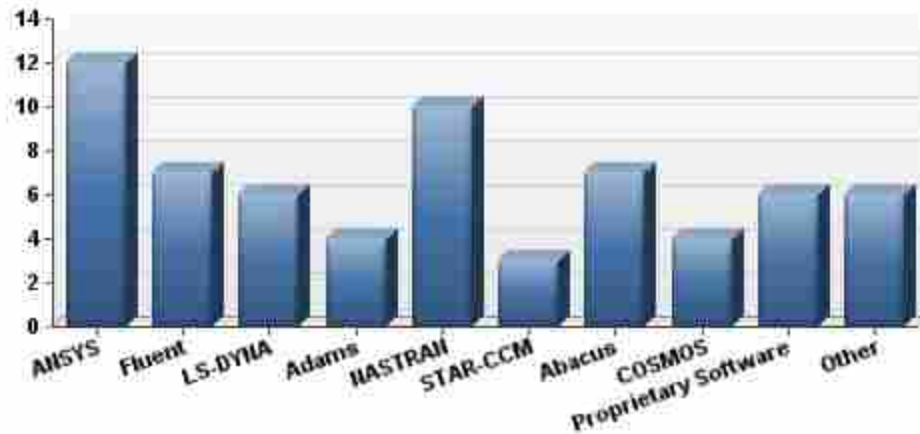
15. CAD applications used by companies: Other: RTT, Inventor, ProCAST



16. CAD systems were rated for collaboration as follows:

- One thought NX very bad: People not understanding the system capability
- One thought SolidWorks very bad: People not understanding the system capability
- One thought Other - ProCAST very bad: People not understanding the system capability
- Five thought NX very good: No Response
- One thought CATIA very good: No Response
- Two thought Pro-E very good: No Response
- One thought SolidWorks very good: No Response
- One thought AutoCAD very good: No Response
- One thought Alias very good: No Response
- Two thought Proprietary Software very good: No Response
- One thought Other very good: No Response

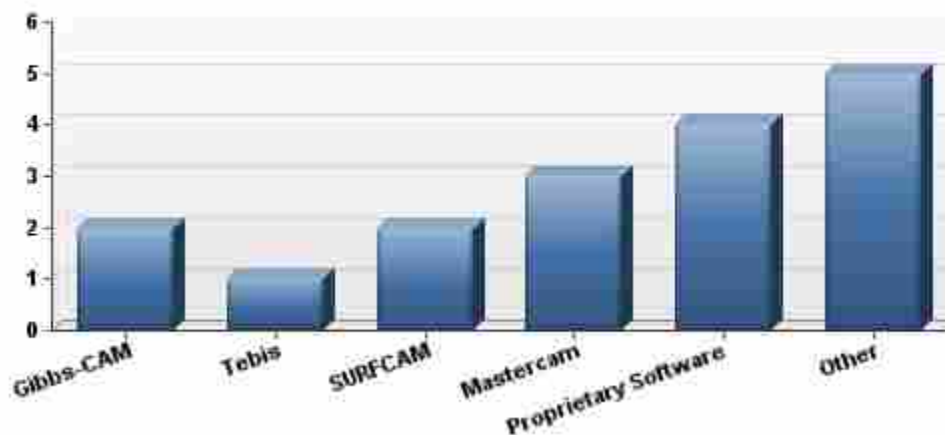
17. Analysis applications used by companies: Other: Many Others; Hypermesh, iSight, MatLab, etc.; Altair HyperWorks; None; Patran; maya, Recurdyn



18. Analysis applications were rated for collaboration as follows:

- One thought ANSYS very good: No Response
- One thought LS-DYNA very bad: Interfacing with loads and inputs.
- One thought NASTRAN very bad: Collaboration with analysis packages is only as good as the viewing and realtime manipulation of the results data. None of the solvers like ABACUS and NASTRAN are good at collaboration other than file transfer.
- One thought NASTRAN very good: No Response
- One thought STAR-CCM very good: No Response
- One thought Abacus very bad: Collaboration with analysis packages is only as good as the viewing and realtime manipulation of the results data. None of the solvers like ABACUS and NASTRAN are good at collaboration other than file transfer.
- One thought Proprietary Software very good: No Response
- One thought Other – None (did not select any analysis packages) very bad: We have no analytical base and that needs to be built in over time.

19. CAM applications used by companies: Other: Tecnomatix; Pro E CAM, NX CAM; NX, CATIA, Pro/E; NX; don't know

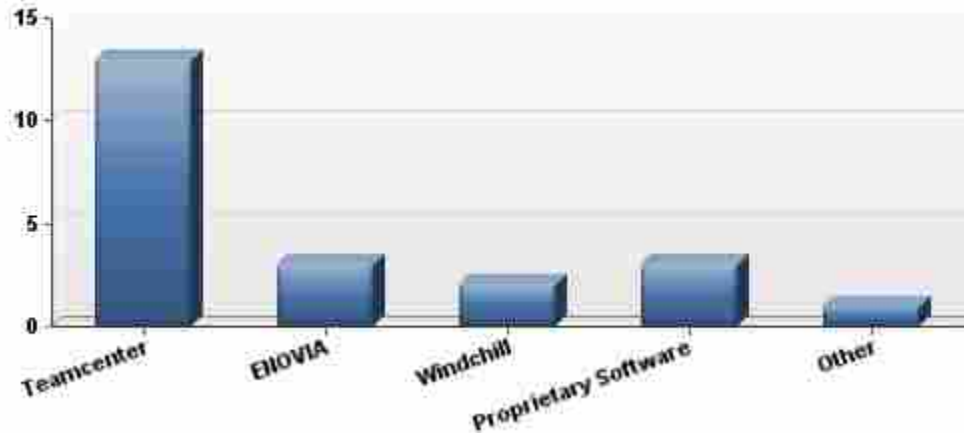


20. CAM applications were rated for collaboration as follows:

- One thought Mastercam very good: No Response

- One thought Proprietary Software very good: No Response
- One thought NX very good: No Response

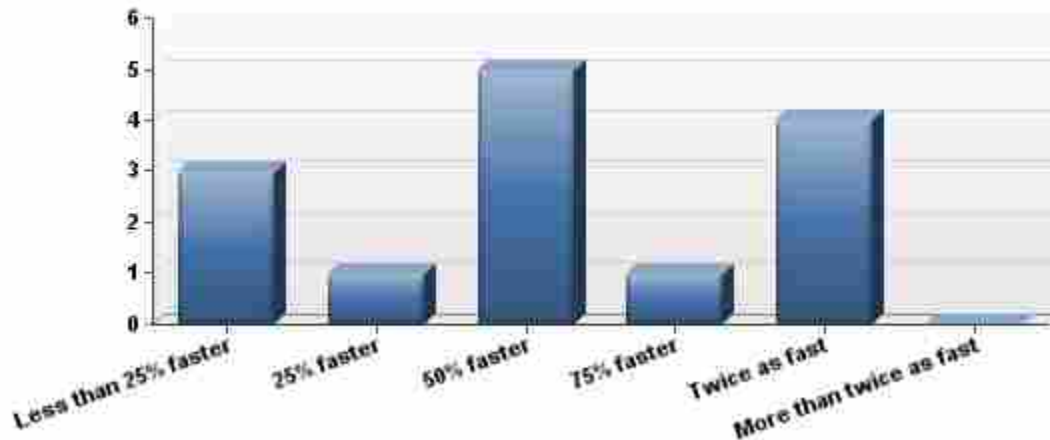
21. PLM applications used by companies: Other: Embedded SW CM



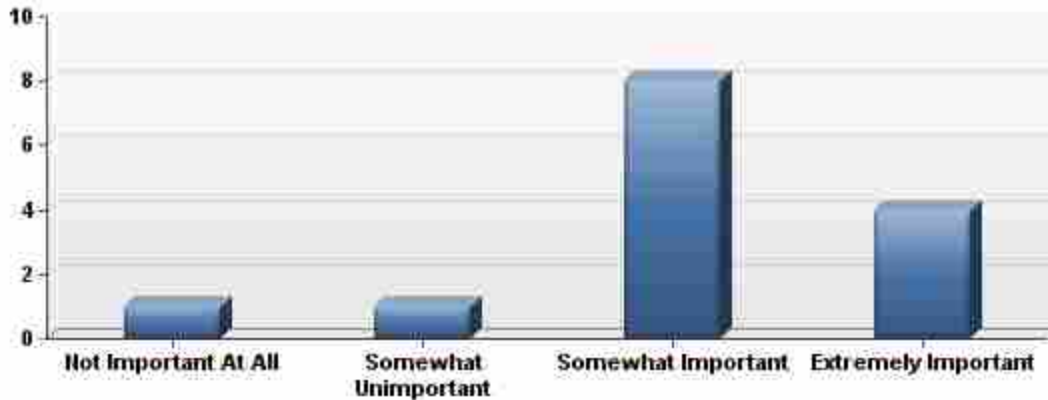
22. PLM applications were rated for collaboration as follows:

- Two thought Teamcenter very good: No Response
- One thought Proprietary Software very bad: Our proprietary s/w codes are usually single decision point based and focused only on one user viewing/interacting.
- One thought Other – Embedded SW CM very good: No Response

23. To benefit the company, v-CAX tools must be faster by: Less than 25% (21%), 25% (7%), 50% (36%), 75% (7%), Twice as fast (29%)



24. Constraints on a multiuser system are: Not Important At All (7%), Somewhat Unimportant (7%), Somewhat Important (57%), Extremely Important (29%)



25. The following is a list of methods for coordinating users in v-CAx Tools and were rated as follows: Very useless (1), Useless (2), Somewhat useless (3), Somewhat useful (4), Useful (5), Very useful (6)

Method	Range of Responses	Most Popular Response	Average
A user seeing a list of tasks each of his or her tasks are dependent on	2 to 6	5 (9 respondents)	4.93
A user seeing a list of tasks dependent on each of his or her tasks	2 to 6	5 (7 respondents)	4.93
A supervisory role allowed to grant permissions, or assign tasks to users	2 to 6	5 (10 respondents)	4.86
Listing tasks assigned to a user, whereby clicking on a task would take user to the area to work on that task	2 to 6	5 (8 respondents)	4.57
Allowing for users to lock geometric features from future editing with notes	2 to 6	5 (7 respondents)	4.50
Requiring permission from original creator to delete or edit a feature	2 to 6	5 (8 respondents)	4.50
Allowing one or many users to be assigned to a single task such as a sketch	2 to 6	5 (7 respondents)	4.50
Locking certain tools from the use of one or more users	2 to 6	5 (5 respondents)	4.21
Permission to work or view only when granted entrance to another user's area	2 to 6	3,4,5,6 (3 respondents)	4.14
Having details of other users' tasks require permissions to be viewed	2 to 6	2,3,4,5 (3 respondents)	3.86
Confining users to work within a predetermined geometric space	2 to 6	3,4 (4 respondents)	3.79

26. Concerns for v-CAx Tools specifically in regards to collaboration included:

- Security management with minimum overhead.
- Scalability; ensuring that the user experience remains robust. Collaboration that is clunky or is error prone with several users will quickly be abandoned by the end users. End user adoption to the collaboration framework is key if the collaboration is going to "stick" and have lasting positive impacts.
- Speed across company sites. It must be responsive.
- Getting them integrated across the entire value stream.
- Security.
- There are unknown-unknowns that will need to be found. We are hoping that beta versions of the tools will be available to ferret out possibly company unique issues or application unique issues sooner than later.

- Electronic workflow, where the correct information is given to the right person(s) at the right time with process inputs/outputs clearly defined. I would also like to see industry move towards having libraries of off the shelf, custom parts/features (I know, it sounds like an oxymoron), i.e. leveraging Knowledge Reuse. Designers are assigned the tasks of creating customized, re-usable, adaptable features/parts. Collaboration comes into play when Designers collaborate on the best way to create the custom feature, i.e. best in class methodologies. There also needs to be a way for Designers and the electronic workflow know when a reusable component is ready for use.
- While some of the access control ideas above are interesting, the nature of rapid development may make these less desirable in practice. When we're really trying to move fast on a project, I envision us wanting our experts to have *more* ability to fluidly move from area to area, helping as needed, rather than less. I could see the access controls quickly getting in the way. Concepts around tracking decisions and associating decision-making with geometry driven by those decisions will be increasingly important in this world. It will be extremely easy for a designer who is new to a team project to step in and start second-guessing the cumulative wisdom of the team to date -- particularly if there is not a very simple way for that new designer to understand which aspects of the design were very deliberately determined, and which are more open for modification.
- Being able to effectively establish design requirements.
- The inefficiencies of work being repeated due to the influence (cause of change) of one area of work on another.
- Real estate claims on ongoing projects.
- Seamless integration with the interface. Data update speed.

27. Company most look forward to the following from v-CAX becoming available:

- Motivating commercial tool vendors to have a framework for integrating tools for collaboration with a rich underlying data model which controls what "role" and perform what "actions".
- From a mentoring aspect, we see significant benefits in allow a "senior" designer to work with or oversee a "junior" designer. Similar to a student driver car with car controls (pedals, steering wheel, etc.) on the passenger side for the instructor. Team work for parallel work to accelerate work at deadlines; getting work out the door at the deadline is something that we spend significant overtime on for the "1 user" who owns the model. Being able to help him would be great.
- Ability to have company personnel and customers work together in an efficient manner to speed up the design process.
- Getting to a fixed process much more rapidly and reducing scrap.
- Getting multiple engineers working together.
- Better collaboration.
- There is the opportunity to change work load scheduling and hand-offs. We can change the priority and people allocations to support those parts of the model that need to be matured to enable complex supporting analyses to be conducted. The parts of the model that are not affecting these analyses can be finalized later.
- Don't know much about it other than what little has been mentioned in this survey, so I really can't comment. We've had concurrent engineering at the product level and we really liked it. I would have to see or think about the benefits of concurrent engineering at the part/feature level. I can see some benefits, but perhaps a bigger benefit would be to develop a system where Designers create reusable features that could be pulled from a library that adapt to its new environment (UDF's, User Defined Features).
- Real-time distributed development of complex models could lead to some very interesting new ways of working. It could be very interesting to allow a distributed team of experts (common in our company) to "gang up on" a complex model over the course of a few days.
- The input and output having a common format with supplier and customer systems.
- Real time concurrent engineering with suppliers.
- Rapid design.