



All Theses and Dissertations

2016-03-01

Understanding and Addressing Collaboration Challenges for the Effective Use of Multi-User CAD

David James French
Brigham Young University

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

 Part of the [Mechanical Engineering Commons](#)

BYU ScholarsArchive Citation

French, David James, "Understanding and Addressing Collaboration Challenges for the Effective Use of Multi-User CAD" (2016). *All Theses and Dissertations*. 6229.

<https://scholarsarchive.byu.edu/etd/6229>

This Dissertation 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.

Understanding and Addressing Collaboration Challenges
for the Effective Use of Multi-User CAD

David James French

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

Doctor of Philosophy

W. Edward Red, Chair
C. Greg Jensen
Sheli Sillito-Walker
John L. Salmon
Kevin B. Tew

Department of Mechanical Engineering

Brigham Young University

March 2016

Copyright © 2016 David James French

All Rights Reserved

ABSTRACT

Understanding and Addressing Collaboration Challenges for the Effective Use of Multi-User CAD

David James French
Department of Mechanical Engineering, BYU
Doctor of Philosophy

Multi-user computer-aided design (CAD) is an emerging technology that promises to facilitate collaboration, enhance product quality, and reduce product development lead times by allowing multiple engineers to work on the same design at the same time. The BYU site of the NSF Center for e-Design has developed advanced multi-user CAD prototypes that have demonstrated the feasibility and value of this technology.

Despite the possibilities that this software opens up for enhanced collaboration, there are now a new variety of challenges and opportunities to understand and address. For multi-user CAD to be used effectively in a modern engineering environment, it is necessary to understand and address both human and technical collaboration challenges. The purpose of this dissertation is to understand and address these challenges.

Two studies were performed to better understand the human side of engineering collaboration: (1) engineers from multiple companies were interviewed to assess the collaboration challenges they experience, and (2) players of the multi-player game Minecraft were surveyed and studied to understand how a multi-user environment affects design collaboration. Methods were also developed to address two important technical challenges in multi-user CAD: (1) a method for detecting undo conflicts, and (2) additional methods for administering data access.

This research addresses some of the important human and technical collaboration challenges in multi-user CAD. It enhances our understanding of collaboration challenges in engineering industry and how multi-user CAD will help address some of those challenges. It also enhances our understanding of how a multi-user design environment will affect design collaboration. The method developed for detecting conflicts that occur during local undo in multi-user CAD can be used to block conflicts from occurring and provide the user with some information about the cause of the conflict so they can collaborate to resolve it. The methods developed for administering data access in multi-user CAD will help protect against unauthorized access to sensitive data.

Keywords: multi-user CAD, synchronous collaboration, engineering collaboration, Minecraft, undo, data access control

ACKNOWLEDGMENTS

I would like to thank my wife, Diantha, for her support of my efforts. Thanks also to my graduate committee for their support and guidance and to Dr. Red, my committee chair. I also appreciate the assistance and support of the many students I have worked with, especially my fellow student co-authors and those who helped directly with my research: Brett Stone, Ammon Hepworth, Tom Nysetvold, Scot Wilcox, Jonathan Tsai, Glen Phelps, Catalina Johnson, Shelby Dushku, Dustin Robinson, and Josh Ruchty. Also a special thank you to the NSF Center for e-Design and to the BYU CADLab's industry sponsors who funded much of this research: Boeing, Pratt and Whitney, Belcan, PCC Airfoils, Spirit AeroSystems and CD-adapco. I am also grateful for the many engineers and Minecraft players who participated in interviews and surveys for this research, and for the support of Jacob Granberry, the WesterosCraft project leader.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
NOMENCLATURE	x
Chapter 1 Introduction	1
1.1 Multi-User CAD	1
1.2 Problem Statement	5
1.2.1 Human Collaboration Challenges in Multi-User CAD	5
1.2.2 Technical Collaboration Challenges in Multi-User CAD	7
1.3 Dissertation Organization	9
Chapter 2 Background	11
2.1 Collaboration Challenges in Engineering Industry	12
2.1.1 Related Research	12
2.1.2 Summary	13
2.2 Multi-User Design Teamwork	14
2.2.1 Related Research	16
2.2.2 Justification of Assumptions	16
2.2.3 Summary	20
2.3 Detection of Local Undo Conflicts in Multi-User CAD	20
2.3.1 Undo Types and Intent	21
2.3.2 Command Pattern	21
2.3.3 Memento Pattern	22
2.3.4 Linear vs. Nonlinear Undo	23
2.3.5 CAD Feature Dependencies	23
2.3.6 Undo Conflicts	25
2.3.7 Syntactic and Semantic Conflicts	25
2.3.8 Related Research	26
2.3.9 Summary	28
2.4 Administering Data Access in Multi-User CAD	28
2.4.1 Access Control Models	29
2.4.2 Related Research	30
2.4.3 Summary	32
Chapter 3 Objectives	33
Chapter 4 A Study of Design Collaboration Challenges in Engineering Industry	35
4.1 Method	35
4.2 Results	35
4.2.1 Demographics	36

4.2.2	Team Size	36
4.2.3	Time Spent Collaborating	37
4.2.4	Effectiveness of Team Collaboration	39
4.2.5	Meetings	43
4.2.6	Virtual Teaming Challenges	46
4.2.7	Organizational Structure	49
4.2.8	Communication Tools	51
4.2.9	Technical Tools	58
4.3	Conclusions	60
4.3.1	Principles	63
Chapter 5 A Study of Multi-User Design Teamwork in Minecraft		64
5.1	Method	64
5.2	Results	64
5.2.1	Team Organization	65
5.2.2	Management	68
5.2.3	Collaboration	72
5.2.4	Comparison of E-Handbook and Minecraft Studies	79
5.3	Conclusions	80
5.3.1	Principles	82
Chapter 6 A Method for Detecting Local Undo Conflicts in Multi-User CAD		83
6.1	Method	83
6.1.1	Classification of Undo Conflicts	83
6.1.2	Method for Detecting Undo Conflicts	87
6.1.3	Successive Undo Commands by Multiple Users	93
6.1.4	Handling Multiple Operations in a Single User Command	93
6.1.5	Redo	93
6.2	Results	95
6.3	Conclusions	95
6.3.1	Principles	96
Chapter 7 Methods for Administering Data Access in Multi-User CAD		97
7.1	Method	97
7.1.1	Administering Spatial Constraints in a CAD Part	97
7.1.2	Restricting Users from Viewing a Spatial Region in a CAD Part	99
7.1.3	Administering Access Using a Discretionary Access Control Model	101
7.2	Results	105
7.2.1	Administering Spatial Constraints in a CAD Part	106
7.2.2	Restricting Users from Viewing a Spatial Region in a CAD Part	107
7.2.3	Administering Access Using a Discretionary Access Control Model	109
7.3	Conclusions	113
7.3.1	Principles	115
Chapter 8 Conclusions		116

8.1	Summary	119
8.2	Limitations and Future Work	119
REFERENCES		121
Appendix A Technical Details of Undo Conflict Detection Method		131
A.1	Data Structure and Acronyms	131
A.2	Method Details	132
A.2.1	Step 1: Before the User Performs a Standard Command	133
A.2.2	Step 2: After the User Performs a Standard Command	133
A.2.3	Step 3: Check for Conflicts	133
A.2.4	Step 4: Handle Detected Conflicts	134
A.2.5	Step 5: After a Successful Undo	134
A.2.6	Checking for Redo Conflicts	135

LIST OF TABLES

2.1	Learning from Minecraft and single-user CAD to understand multi-user CAD . . .	16
2.2	Multi-user CAD permission types and researchers who have studied each type . . .	31
4.1	Effectiveness and typical length of formal and ad-hoc meetings	45
5.1	Comparison of findings from the E-Handbook and Minecraft studies	80
6.1	Pre-operation and post-operation feature states	84
6.2	Implications and suggested system responses for Feature-Self (Type A) undo conflicts	89
6.3	Implications and suggested system responses for Feature-Child (Type B) and Feature-Parent (Type C) undo conflicts	90

LIST OF FIGURES

1.1	Collaborative engineering environment in 1957	1
1.2	One person controlling a CAD model while others observe	2
1.3	Students collaborate using BYU's NXConnect multi-user CAD prototype	3
2.1	Minecraft players working in a shared multi-user environment	14
2.2	A complex structure built by the WesterosCraft team in Minecraft	15
2.3	Operation history list	22
2.4	An example of a command object	22
2.5	A feature references its parent feature	24
2.6	Child features of a sketch feature	24
2.7	Parent features of an extrude feature	24
2.8	Example of a syntactic conflict	26
2.9	Example of a semantic conflict	27
3.1	Some important human and technical design collaboration challenges in multi-user CAD	34
4.1	Education levels of study participants	36
4.2	Histogram of team sizes	37
4.3	Percent of time spent collaborating	38
4.4	Percent of time spent collaborating by role	38
4.5	Team collaboration effectiveness ratings	39
4.6	What makes team collaboration effective	40
4.7	What makes team collaboration ineffective	40
4.8	Barriers to collaboration	42
4.9	Percent of time spent in formal meetings	44
4.10	Percent of time spent in meetings by role	45
4.11	Familiarity with local vs. virtual collaborators	50
4.12	Communication tools	52
4.13	Number of emails received per day	53
4.14	Average number of emails received per day by role	54
4.15	Percent of time spent on the phone	55
5.1	Size of the best team respondents have worked with in Minecraft	66
5.2	Desired teammate attributes	67
5.3	Techniques used by Minecraft teams to divide work	71
5.4	How Minecraft players typically communicate with their teammates	74
5.5	Average effectiveness ratings of communication tools	75
5.6	Which communication tools participants would like integrated into Minecraft	75
5.7	Concurrency of teamwork in Minecraft	77
5.8	Effectiveness of learning methods in Minecraft	78
5.9	Teamwork challenges in Minecraft	79
6.1	Standard and undo commands	84

6.2	Example of a Feature-Self (Type A) undo conflict	85
6.3	The four subtypes of Feature-Self (Type A) undo conflicts	85
6.4	Example of a Feature-Child (Type B) undo conflict	86
6.5	The four subtypes of Feature-Child (Type B) undo conflicts	87
6.6	Example of a Feature-Parent (Type C) undo conflict	88
6.7	The four subtypes of Feature-Parent (Type C) undo conflicts	88
6.8	Operation history list	90
6.9	Each command stores an ordered list of operations	91
6.10	Example of successive undo commands	94
7.1	Spatial constraints in a partial R/C car CAD model	100
7.2	Constraint surface hide/show options	101
7.3	NXConnect spatial constraint database table	101
7.4	Spatial constraints administration form	102
7.5	DAC part access administration form	104
7.6	NXConnect Parts database table definition	105
7.7	A manager creates three spatial constraints	107
7.8	User views after being assigned to spatial constraints	108
7.9	Four users' views after being assigned spatial constraint restrictions with the "Hide restricted objects" option enabled	109
7.10	The manager of a CAD part assigns other users access to contribute to and view that part using discretionary access control	110
7.11	An NXConnect error message warns the manager of a CAD part that each part must have at least one Manager	111
7.12	A non-manager of a CAD part is blocked from accessing the spatial constraints administration form	111
7.13	A non-manager can view the part access settings but cannot administer them	112
7.14	A user with view access opens a CAD model	113
7.15	A user assigned view-only access to a part is blocked from editing	114
7.16	A user without view access to a part is blocked from opening it	115
A.1	Data stored as attributes of CAD features	131
A.2	Data stored in a command in the undo list	132
A.3	Data sent to other users	132

NOMENCLATURE

Terms and Definitions

Feature	The CAD building block unit consisting of controlling parameters in feature-based CAD which are combined to create a parametric geometry model
Semantic conflict	Conflicts which occur when multiple users perform operations which violate the design intent of the model, leading to process waste
Syntactic conflict	Conflicts which cause inconsistency or an invalid state of the model, causing software bugs

Acronyms

CAX	Computer-aided [design, engineering, manufacturing, etc.]
CAD	Computer-aided design
PLM	Product lifecycle management
PDM	Product data management
API	Application programming interface
EAR	Export Administration Regulations
ITAR	International Traffic in Arms Regulations
OEM	Original equipment manufacturer

CHAPTER 1. INTRODUCTION

1.1 Multi-User CAD

Decades ago, engineers worked at drafting tables like the one in Fig. 1.1 in a synchronous collaborative environment. Everyone had their own view and their own pencil. Not only was it easy to consult with other engineers, it was also possible to work on the same drawing at the same time. If there was a drawing that was a bottleneck in the process, you could simply ask more engineers to work on it to get it done faster. Everyone could see what everyone else was working on, and how fast each individual's work was progressing. Supervisors and experts also had the opportunity to discover and correct errors early in the design process. Design consensus occurred naturally because designers and engineers could observe each others' work throughout the design process.



Figure 1.1: Collaborative engineering environment in 1957 [1].

Over the last few decades, computer-aided design (CAD) applications have become mainstream in engineering industry. CAD programs have become the lifeblood of almost every engineering effort. After a concept is developed, a CAD model is created. The CAD model is then used by many other engineering teams for a wide range of activities, from design analysis and manufacturing planning to product visualization for marketing.

Although CAD automates many design tasks, it was built with the same mindset as many other software applications developed during the 1970's through 1990's: it was built for one user to work on one file at a time, where the file represents a CAD part or model. Collaborators are restricted to an observer's role or to sharing control of a single mouse cursor through a screen-sharing application (see Fig. 1.2).



Figure 1.2: One person controlling a CAD model while others observe.

Multi-user CAD is an emerging technology that promises to facilitate collaboration, enhance product quality, and reduce product development lead times by allowing multiple engineers to work on the same CAD model at the same time [2,3]. Multi-user software describes a networked software application that allows multiple users to view and modify model data at the same time, rather than have a single user own and modify a file and then transfer that file to another user.

The BYU site of the NSF Center for e-Design has developed advanced multi-user CAD prototypes that have demonstrated the feasibility and value of this technology. Figure 1.3 shows

four BYU students working together to design an unmanned aerial vehicle (UAV) wing using NXConnect. NXConnect is one of BYU's multi-user CAD prototypes and is a plugin for Siemens' NX that adds multi-user functionality to NX.



Figure 1.3: Students collaborate using BYU's NXConnect multi-user CAD prototype.

Some multi-user CAD environments are reported in the literature, including webSpiff [4], TOBACO [5], RCCS [6], COCADCAM [7], CollabCAD [8], and others [9–13]. Until recently none had offered both a multi-user experience and commercial or near-commercial-level functionality [3]. Recent research has produced an architecture and implementation that offers persistent naming of model entities (e.g. features) [14], data consistency [2, 15, 16], support for commercial file formats [17], integrated collaborative tools [14, 18], and a general vision for future development [3]. Evaluation of this multi-user CAD system in academic settings and by industrial research sponsors is ongoing.

Multi-user applications in other industries besides engineering are becoming more widespread. Multiplayer online games have existed for years. Word processing and spreadsheet programs have also begun to support multi-user capabilities: Google [19], Microsoft [20], and Apple [21] all now offer multi-user word processing and spreadsheet software.

Engineering applications, which are typically more complex and computationally intense than consumer applications, have been slower to follow that trend. One online CAD product called Onshape appeared in public beta in March 2015 which supports multi-user design [22, 23]. Onshape currently competes for use by small and mid-size companies in roughly the same market as Dassault SolidWorks, Siemens Solid Edge, and Autodesk Fusion 360 [24]; it is not yet commonly used by large engineering companies.

Multi-user functionality is worth researching because it may provide the following benefits:

- Enhance team awareness.
- Help catch mistakes sooner.
- Provide more opportunities for collaborative problem-solving.
- Reduce design cycle time through greater parallelization of bottleneck tasks.
- Shorter feedback loops.
- Enhance mentoring.
- Simplify version control.
- Make design cycle times more predictable.
- Server-side data backups.

Multi-user software may also affect the following aspects of design collaboration:

- Planning, organization, and management
- Communication
- Data access control and security

1.2 Problem Statement

Despite the possibilities that multi-user CAD software opens up for enhanced collaboration, there are now a new variety of challenges and opportunities to understand and address before multi-user CAD technology can be used effectively [3]. Wolter et al. claim that “digital tools for collaboration introduce new kinds of problems that need to be addressed,” [25]. They then describe two factors for successful collaborative engineering: “To utilize all the opportunities. . . the collaborative engineering processes, methods and tools can not only focus on the *technical* goals and how to achieve them, they must also consider the *human* factor. Only then the task work can be united with teamwork for successful collaboration,” (emphasis added) [25]. Thus, for multi-user CAD to be used effectively in engineering industry, it is necessary to understand and address both human and technical collaboration challenges. The purpose of this dissertation is to understand and address these challenges.

1.2.1 Human Collaboration Challenges in Multi-User CAD

Human collaboration is a critical part of the product design process in modern engineering industry. Teams combine the skills, experience, and expertise of their members to create products that a single person could never make. Modern technology has automated many aspects of the design process and allowed nearly instantaneous cross-continent communication. But human collaboration remains, and will probably always remain, one of the great enablers of large engineering projects. Here are two important questions about human collaboration in multi-user CAD:

1. What collaboration challenges are experienced in engineering industry, and what can we learn how will multi-user CAD help address some of those challenges?
2. How does a multi-user design environment affect design collaboration?

Collaboration Challenges in Engineering Industry

The first question posed above in regards to understanding human collaboration in multi-user CAD is, “What collaboration challenges are experienced in engineering industry, and how will multi-user CAD help address some of those challenges?” As the engineering process becomes

more distributed across the globe, collaboration becomes more complex [26, 27], and there are many factors that influence the effectiveness of collaboration. A study of collaboration challenges in engineering industry may provide some insight into how multi-user CAD will help address some of those challenges.

Impact of Multi-User CAD on Design Collaboration

The second question posed above in regards to understanding human collaboration in multi-user CAD is, “How does a multi-user design environment affect design collaboration?” Specifically:

- How will multi-user teams be organized?
- How will they plan?
- How will they perform task management?
- How will they communicate?
- How much of the time will they work concurrently (at the same time) in the same model?
- How will the multi-user capability affect mentoring and training?

These questions could be very difficult to answer before multi-user CAD is adopted by large engineering organizations [28]. Since commercial multi-user CAD applications are not yet widely used by large engineering organizations, are there any other multi-user design applications that we can use to better understand multi-user design?

Minecraft is a popular online multiplayer game in which virtual teams of up to forty designers can collaborate in real-time to build amazing structures using blocks [29–31]. It is a natural experiment in collaborative 3D design of complex structures. Virtual teams of up to forty simultaneous designers have created city-scale models with total design times in the thousands of hours [30].

Although there are significant differences between Minecraft and multi-user CAD, there are some compelling similarities as well. For example, both CAD and Minecraft are used to create

3D virtual designs, both require strong design skills, and both can involve lots of planning and communication using distributed teams. Cody Sumter of the MIT Media Lab has implied that there are some similarities between building in Minecraft and designing in CAD [32].

Studying multi-user design in Minecraft may provide insight into how a multi-user design environment will affect design collaboration, including the effects of multi-user CAD on team organization, planning, design concurrency, communication, and mentoring.

1.2.2 Technical Collaboration Challenges in Multi-User CAD

The second type of collaboration challenge in multi-user CAD is the technical type. The technical side of multi-user CAD collaboration includes the methods, algorithms, and experience that the software provides to users to facilitate and enhance their collaboration.

Many technical challenges in multi-user CAD have already been addressed: Jing et al. and Liao et al. have studied identification of CAD features [33, 34]; Hepworth and Moncur have studied feature reservation for conflict avoidance [16, 35]; and Hepworth et al. and Xiantao et al. have studied client model consistency [15, 36].

One engineering company collected data on which buttons were clicked most in single-user CAD, and they reported to BYU that in a data set of about 13,000 button clicks, undo constituted about 18% of all button clicks—more clicks than any other individual button. Undo was clicked about the same amount as “delete” and “finish sketch,” and about ten times more than the most frequently-clicked feature (edge blend). Since undo is such a popular function in single-user CAD, we expect it to be a popular and important function in multi-user CAD as well.

Now, the mechanism that supports undo in single-user CAD is roughly equivalent to what is known as “global undo” in multi-user CAD, since multiple users are essentially acting as one user to undo their commands; however, the intent of single-user CAD—to undo the local user’s command—is best supported by what is known as “local undo.” Local undo means to undo the local user’s last command. Unfortunately, local undo can conflict with other users’ more recent commands. Thus, a method is needed to detect and prevent local undo conflicts in multi-user CAD.

Data access control is an important part of collaboration between individuals and companies in engineering industry since CAD designs can, and often do, contain critical proprietary data [37]. Many engineering companies are particularly sensitive to the security of the intellec-

tual property contained in their CAD data, since much of their data is proprietary, and since some data are subject to national regulations such as Export Administration Regulations (EAR) [38] and The International Traffic in Arms Regulations (ITAR) [39]. Multi-user CAD applications need to incorporate methods to protect design data to prevent loss of valuable proprietary data. Since by nature a multi-user CAD program exercises a high degree of control over the CAD system, it should be able to restrict certain user actions while retaining support for multi-user collaboration in a full-featured CAD model.

Teng et al. studied a hybrid Mandatory Access Control (MAC) and Role Based Access Control (RBAC) access model for multi-user CAD part files [40]. Moncur et al. studied methods for restricting permissions to edit and view features [16,41]. Red et al. and Marshall et al. studied spatial constraints for partitioning a model in multi-user CAD [42,43]. Administering data access control is an important topic that has received significant attention to date, and due to its critical nature in sensitive engineering industries, it deserves even more attention.

Considering the previous technical research that has been conducted on multi-user CAD, the following technical challenges should be addressed next to facilitate collaboration in and industry adoption of multi-user CAD:

1. How can undo conflicts be detected and blocked in multi-user CAD?
2. What are some additional methods for administering and restricting data access in multi-user CAD?

Detection of Local Undo Conflicts in Multi-User CAD

The first question posed above in regards to addressing technical challenges in multi-user CAD is, “How can undo conflicts be detected and blocked in multi-user CAD?” Undo/redo poses a technical challenge in multi-user CAD that is not an issue in single-user CAD, assuming the intent of an undo or redo command is to undo the most recent command of the local user (most other multi-user applications treat undo in this way).

Both syntactic and semantic conflicts can occur during multi-user undo [44]. And due to the high number and complexity of dependencies between CAD features, these conflicts are more

likely to occur and are more difficult to handle than conflicts in many other types of multi-user software [45].

A classification of the types of conflicts that can occur during the use of local undo in multi-user CAD needs to be developed, as well as a method for detecting and blocking those conflicts. This method can then be used to prevent conflicts, to warn users about those conflicts, and to provide users with information about the cause of the conflict so they can collaborate to resolve it.

Administering Data Access in Multi-User CAD

The second question posed above in regards to addressing technical challenges in multi-user CAD is, “What are some additional methods for administering and restricting data access in multi-user CAD?” More work is being outsourced between companies and even countries as engineering designs become more complex. Multi-user CAD applications need to incorporate methods for protecting design data to prevent loss of valuable proprietary data and comply with government regulations such as Export Administration Regulations (EAR) and International Traffic in Arms Regulations (ITAR) [38, 39].

Since a multi-user program has a high degree of control over the CAD system, it should be able to restrict certain user actions while retaining the support for multi-user collaboration in a full-featured model. For example, if a large U.S. OEM needed to outsource a portion of a certain design to a supplier in Europe using multi-user CAD, they would want a method to allow the foreign supplier to access only the necessary portions of the design. The supplier should only be able to view and edit what is necessary to accomplish their assigned tasks, while restricting access to other portions of the design. Such methods could utilize hiding or blocking access to specific regions or models to allow for appropriate collaboration to occur both inside and outside of an organization while protecting sensitive data.

1.3 Dissertation Organization

The organization of this dissertation is as follows: Chapter 1 introduces the reader to multi-user CAD and presents the problem statement which includes four important research questions which will be investigated in this dissertation. Chapter 2 presents related research and background

context related to these research questions. Chapter 3 defines the four main research objectives of this dissertation in greater detail. Chapters 4 through 7 each explain the methods, results, and conclusions used to accomplish one of the four main research objectives. Chapter 8 presents overall conclusions and suggestions for future research.

CHAPTER 2. BACKGROUND

Despite the possibilities that multi-user CAD software opens up for enhanced collaboration, there are now a new variety of challenges and opportunities to understand and address before multi-user CAD technology can be used effectively [3]. Wolter et al. describe two factors for successful collaborative engineering: “To utilize all the opportunities. . . the collaborative engineering processes, methods and tools can not only focus on the *technical* goals and how to achieve them, they must also consider the *human* factor. Only then the task work can be united with teamwork for successful collaboration,” (emphasis added) [25]. Thus, for multi-user CAD to be used effectively in a modern engineering environment, it is necessary to understand and address both human and technical collaboration challenges.

As mentioned previously in section 1.2.1, here are two important questions about human collaboration in multi-user CAD that need to be answered:

1. What collaboration challenges are experienced in engineering industry, and how will multi-user CAD help address some of those challenges?
2. How does a multi-user design environment affect design collaboration?

Sections 2.1 and 2.2, below, discuss background context and research related to these questions.

The second type of collaboration challenge in multi-user CAD is the technical type. The technical side of multi-user CAD collaboration includes the methods, algorithms, and experience that the software provides to users to facilitate and enhance their collaboration.

As mentioned previously in section 1.2.2, here are two important technical challenges in multi-user CAD that need to be addressed to facilitate collaboration in multi-user CAD:

1. How can undo conflicts be detected and blocked in multi-user CAD?

2. What are some additional methods for administering and restricting data access in multi-user CAD?

Sections 2.3 and 2.4, below, discuss background context and research related to these challenges.

2.1 Collaboration Challenges in Engineering Industry

2.1.1 Related Research

Engineers throughout the world collaborate extensively with each other to share expertise and work towards common goals. Although many studies explore team and organizational effectiveness in general terms, there is very little information available about project team collaboration in the engineering industry.

Ovum IT, a research and advisory firm, conducted a study in 2014 on physical and virtual meeting behaviors and the evolving requirements of the collaborative workforce for virtual collaboration and web conferencing [46]. They found that more than 50% of employees claimed that the number of meetings they have is increasing, and 67% of employees reported that more than half of the meetings they attend are not of value. “Yet, few workers would argue that communication and collaboration are critical to achieving their goals.” They also found that more than a third of all meetings are ad hoc, that 32% of all meetings are virtual, and that younger workers (ages 26-35) hold 38% of their meetings virtually.

Ovum’s research also reported an increasing trend of using mobile devices for virtual meeting collaboration. They also reported frustration with traditional web conferencing tools, and late start times cited as a key reason that meetings are perceived to fail to deliver value. It will be valuable to capture how engineering industry compares in some of these metrics.

Engineering collaboration is unique due to its heavy use of complex 3D design data, its unique toolset, its deeply hierarchical team structure, and its strict security constraints [25, 26, 47]. Very few studies have surveyed practical collaboration challenges in engineering industry. Wolter et al. describe a brief study of problems and questions in engineering collaboration [25]. They emphasize certain needs and difficulties in collaborative engineering such as the need for instant collaborative sessions, the difficulty of implementing collaborative solutions in a heterogeneous IT

landscape, and the difficulty of protecting intellectual property (IP) rights. They also emphasize the need to consider human factors in collaboration. They even recognize that “the concept of screensharing lacks the possibility of an equal participation in the collaboration scenario, because only one participant can present his screen to all others,” which argues the need for equal multi-user participation.

Müller et al. conducted a study on collaborative product development and digital engineering tools in German industry in 2011 [47]. The data was gathered via an electronic survey with 1401 responses from engineers. Müller et al. found that 64% of those engineers said that communication, collaboration, and negotiation takes up more than 20% of their working time. They claim that “efficient collaboration amongst companies and disciplines is a key for successful engineering of complex systems.” This study comes the closest to studying engineering collaboration in the way it needs to be studied; and its findings highlight the need to understand engineering collaboration better than we currently do. A similar and more thorough study needs to be performed to provide further insight into modern design collaboration challenges in industry. In-person interviews with engineers, and a broader and deeper set of collaboration questions would add significant value to such a study.

A study of modern design collaboration practices and challenges in engineering industry will help us understand those challenges better and help us understand how industry is currently dealing with those challenges. It will reveal how multi-user CAD may address some of those challenges. This will allow us to better define the requirements of a multi-user CAD application, to understand the future role of multi-user CAD in engineering industry, and help us identify principles for using multi-user CAD more effectively in industry.

2.1.2 Summary

Very little research has been conducted on design collaboration practices and challenges in modern engineering industry. A study of modern design collaboration practices and challenges in engineering industry will be very valuable for understanding the future role of multi-user CAD in engineering industry and for identifying principles for using multi-user CAD effectively in industry.

2.2 Multi-User Design Teamwork

Minecraft is a popular online multiplayer game in which virtual teams of up to forty designers can collaborate in real-time to build amazing structures using blocks [29–31]. Figure 2.1 shows four players playing Minecraft together.



Figure 2.1: Minecraft players working in a shared multi-user environment.

Although there are significant differences between Minecraft and multi-user CAD, there are some compelling similarities as well. For example, both CAD and Minecraft are used to create 3D virtual designs, both require strong design skills, and both can involve lots of planning and communication using distributed teams. In fact, Cody Sumter of the MIT Media Lab has said that “[The creator of Minecraft has] tricked 40 million people into learning to use a CAD program,” [32].

The community of nearly 14 million Minecraft users who play the desktop version (not counting Xbox and iOS) represents a vast body of 3D collaborative design experience [29]. Virtual teams of up to forty simultaneous designers have created city-scale models with total design times in the thousands of hours [30]. One famous Minecraft design was produced over a period of years with dozens of designers often working together simultaneously. It represents the entire fantasy world of George R. R. Martin’s *Game of Thrones* [30]. Its virtual landscape is about 1/100 the size of the South American continent [48].

One of the many complex structures built by WesterosCraft is shown in Fig. 2.2. These designers have dealt extensively with issues of communication and coordination within all-virtual teams working on complex 3D design projects, as will be needed in multi-user CAD applications.



Figure 2.2: A complex structure built by the WesterosCraft team in Minecraft (used with permission).

Considering the present difficulty of performing realistic experiments on the effects of multi-user CAD, and considering the similarities between CAD and Minecraft, it follows that Minecraft could be used to study the effects of multi-user CAD on design collaboration, including team organization, planning, design concurrency, communication, and mentoring. These effects could otherwise be very difficult to study before multi-user CAD tools become commonly available [28]. Even once multi-user CAD becomes commonly available, an ideal evaluation of it would require many experienced professionals working on industry-grade projects.

If we can study the expert use of multi-user design applications before multi-user CAD becomes commonly used, then we can better understand how a multi-user environment affects design collaboration.

2.2.1 Related Research

Cody Sumter of the MIT Media Lab has implied that there are some similarities between building in Minecraft and designing in CAD [32]. To my knowledge no one has studied Minecraft to learn about collaborative design or performed similar research.

2.2.2 Justification of Assumptions

If multi-user design in Minecraft is to be used to make predictions about multi-user CAD, it is necessary to justify the assumptions these predictions are based upon. By comparing Minecraft (which is inherently multi-user) to the multi-user CAD systems of the future, it will be easier anticipate the benefits of multi-user CAD. This claim is strengthened by considering the inverse comparison (comparing single-user CAD, which does exist, to single-user Minecraft which does not exist) and postulating how the design experience in Minecraft would be worsened by this change.

Two assumptions which must be examined in regard to these claims are: (1) that Minecraft is similar enough to CAD in purpose and functionality (disregarding obvious multi-user-related differences) and (2) that multi-user process effects are similar enough in both CAD and Minecraft (see Tab. 2.1). These assumptions are examined and justified below.

Table 2.1: Learning from Minecraft and single-user CAD to understand multi-user CAD.

	Minecraft	CAD
Single-User	Though experiment	Current industry standard
Multi-User	Commonly played game	<i>Research focus</i>

Justification: Minecraft and CAD Are Similar in Purpose and Functionality

The justification that Minecraft and CAD are similar enough appears to be the more critical of the two previously mentioned assumptions: Are Minecraft and CAD similar enough, aside from

obvious single-user/multi-user differences, that it is worth attempting to understand how multi-user CAD will be different from contemporary, single-user CAD, based on differences observed between single-user CAD and multi-user Minecraft? Here are some of the similarities between Minecraft and CAD (disregarding obvious single-user/multi-user differences):

- Both tools allow users to create, modify, and delete 3D geometry.
- Both tools can be used to design complex, virtual 3D models.
- Both tools are often used by large, distributed teams.
- Collaboration tools and communication skills are required.
- Extensive planning and organization is often required.
- Leadership, management, and permissions are used in conjunction with these tools.
- Both tools provide high levels of flexibility in how they can be used.
- Both tools have users with varying levels of experience and skill level.

And here are some of the differences between Minecraft and CAD (disregarding obvious single-user/multi-user differences):

- The purpose for which the tool is used.
 - Creative Mode Minecraft is generally used to build virtual models of aesthetic structures for recreation.
 - CAD is used to design and represent virtual models for real products.
- Model data.
 - Minecraft builds (models) consist of voxels (3D pixels) with varying locations, appearances, and functions.
 - CAD design interfaces and models are more complex and consist of more extensive geometric forms, relationships and metadata such as material properties and geometric dimensioning and tolerancing (GD&T).

- User skill types.
 - Minecraft users employ general artistic skill, with more flexible requirements and specifications.
 - CAD users design according to carefully constructed product requirements and specifications, applying specialized product design skills and engineering knowledge.
- User motivations.
 - Minecraft users generally play for fun, reputation, and/or socialization.
 - CAD users generally work for monetary compensation, reputation, as well as potential personal fulfillment.
- Education/experience.
 - Minecraft users don't have formal Minecraft-specific education.
 - Most CAD users in the engineering industry have at least some formal postsecondary training.
- Security/permissions.
 - The security surrounding access to and control of CAD models in much of engineering industry is much greater than in Minecraft, albeit Minecraft administrators do have the ability to impose role and model restrictions.
- Undo.
 - Undo is a very important function in CAD; however, Minecraft does not natively support undo functionality [49], although some plugins are available that local undo functionality to Minecraft [50, 51].

Thus there are significant similarities and differences between Minecraft and CAD (ignoring obvious single-user/multi-user effects). Based on the similarities listed above, Minecraft and CAD are similar enough that it is worth understanding how multi-user CAD will differ from contemporary CAD based on differences observed between single-user CAD and multi-user Minecraft.

Justification that Multi-User Effects are Similar in Minecraft and CAD

Justification also needs to be provided for the second assumption—that multi-user functionality has similar effects in both Minecraft and CAD. A thought experiment can help support this assumption and illustrate the potential effects of a multi-user architecture. Since single-user CAD design processes are well known, the following question can be asked: “What if Minecraft was single-user? How would it be used, and what would be the effects of this change?” Refer to Table 2.1. If the answer reflects on how contemporary single-user CAD is used in industry, then this second assumption has some support.

If Minecraft were single-user, then a lead user would need to create a top down assembly model to organize the various sub-models distributed between users. After each sub-model was completed the lead would be required to integrate the sub-models into the assembly model. Interfaces would be carefully defined and even preliminarily modeled during the planning phase so that users could later integrate their models. Effects of a single-user architecture in Minecraft might include the following:

- It would take significantly more time to plan, manage, and integrate the interfaces to combine the sub-models into an assembly.
- Only one person at a time could integrate sub-models into the assembly model creating a bottleneck and making it difficult for others to contribute and offer input as to how the sub-model should be integrated into the primary model.
- It would be difficult for everyone, including project leaders, to track and understand the progress of the project.
- Changes that occur after the build has begun would be much more expensive (in terms of man-hours), because users would need to be notified and briefed about design changes instead of simply seeing the changes as they appear in the model.
- The quality, coordination, and harmony of the resulting design would decrease because everyone would work on their sub-models based on their individual knowledge, ideas, and styles instead of receiving immediate feedback from other users.

- Users would not catch each other’s mistakes until the models are integrated into the assembly model, requiring timely rework.
- If no data management system was employed, significant time would be required to send files back and forth between users.

The effects described above reflect with reasonable accuracy the effects currently felt in the use of single-user CAD in engineering industry, based on the authors’ experience and as described by David in his research of engineering Product Lifecycle Management systems [52]. This thought experiment therefore supports the assumption that multi-user effects are similar in both Minecraft and CAD.

2.2.3 Summary

No one has conducted a formal study of other multi-user tools such as Minecraft to postulate how multi-user CAD will be used in engineering industry. By studying multi-user Minecraft, we will better understand the principles of multi-user design and thus enhance our ability to use multi-user CAD effectively.

2.3 Detection of Local Undo Conflicts in Multi-User CAD

Undo is a useful and widely supported command that is available in many software applications. A 1976 research report by Miller and Thomas of IBM claimed that “It would be quite useful to permit users to ‘take back’ at least the immediately preceding command (by issuing some special ‘undo’ command),” [53, 54]. Yang asserted, “Most sophisticated interface systems should be provided with an undo support facility,” [55].

Abowd and Dix claimed that “Few people would argue about the importance of undo,” [56]. A large engineering company—one of the BYU CADLab’s research sponsors—recently used an analytics tool provided by BYU to track which buttons their designers clicked the most in single-user CAD. They reported to BYU that the Undo button was clicked about four times more frequently than any other button in their single-user CAD tool (Siemens NX). This makes sense since almost every action in a CAD program has an undo function, and it indicates that undo will be important in multi-user CAD as well.

2.3.1 Undo Types and Intent

According to Wikipedia, “When multiple users can edit the same document simultaneously, a multi-user undo is needed,” [54]. Abowd and Dix describe two types of undo in multi-user applications: (1) local undo, where the local user’s actions are reverted, and (2) global undo, where the actions of all users are undone [56]. Prakash and Knister describe other types of undo as well, such as regional undo or selective (arbitrary) undo [57].

Yang stated that “undo should be seen as an intention of the user, not an aspect of system functionality,” [55]. Chen [58] claims that single-user undo is most commonly used to accomplish the following user intentions:

1. recover from an unintended or incorrect operation,
2. learn a new feature by trial and error, or
3. explore alternatives.

Prakash and Knister found that users typically use local undo more than global undo in multi-user software, since they are typically not aware of the last global action [57]. They found that global undo was primarily useful when the group as a whole wished to go to a known earlier state of a document or when the group was working with synchronized views (effectively working as one person). Based on this finding, it is expected that local undo will be used more frequently in multi-user CAD than any other type of undo; therefore, this dissertation focuses on local undo.

2.3.2 Command Pattern

The Command Pattern is a software design pattern that is very useful for supporting undo in both single-user and multi-user CAD. In the Command Pattern, a historical list of all undoable commands performed on the model is stored [59]. A representation of a command history list is shown in Fig. 2.3. In this figure, Q_i represents the state of the CAD model at a certain time, and c_i represents a specific command object. According to Meyer, “A command object represents the information needed to execute a user-requested operation and, if undoing is supported, cancel it,” [60]. In multi-user CAD, the information needed to execute a user-requested operation is

typically the information about the new state of a certain CAD feature, i.e. Q_i . An example command object is shown in Fig. 2.4.

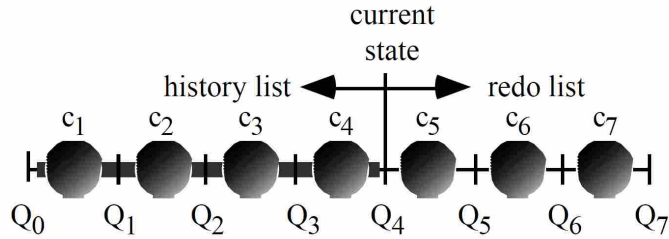


Figure 2.3: Operation history list (from Berlage [44]).

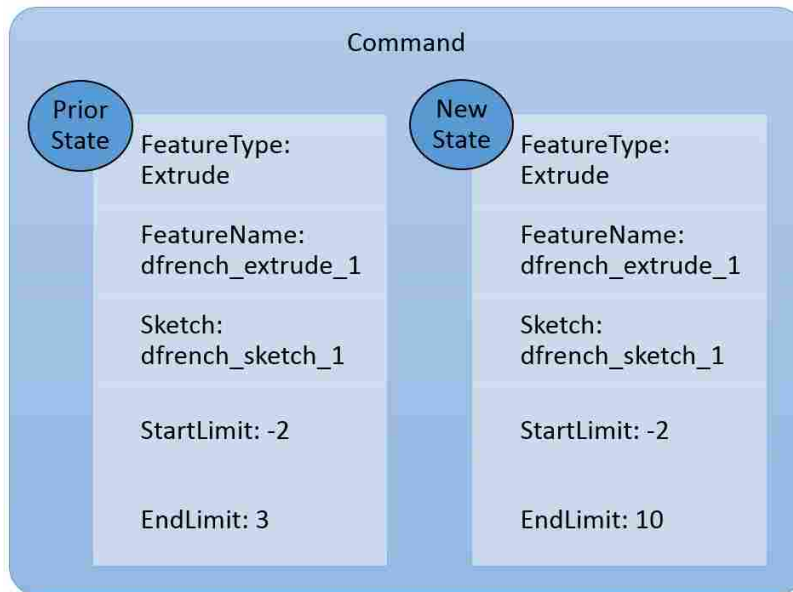


Figure 2.4: An example of a command object.

2.3.3 Memento Pattern

The Memento pattern is another software design pattern that augments the Command pattern by also storing the previous state of a specific CAD feature within each command to make undoing a command easier and more reliable [54, 61, 62]. Thus each command object stores two states: (1) the state of the CAD feature before the command was performed, and (2) the state of

the CAD feature after the command was performed. The command object shown in Fig. 2.4 is an example of a command object that implements the Memento pattern.

Referring back to Fig. 2.3, command c_i stores the information necessary to represent states Q_{i-1} (the prior state) and Q_i (the new state). When c_i is performed, the state Q_i is established in the model. When c_i is undone, the state Q_{i-1} is restored.

2.3.4 Linear vs. Nonlinear Undo

Berlage states that “undo models can be divided into two categories: linear undo, which uses a chain of commands where only the previous and next command can be undone or redone; and nonlinear undo, which allows the user to undo arbitrary actions,” [63]. A command history list is represented in Fig. 2.3, where Q_i represents a certain state, and c_i represents a user command. In linear undo, actions must be undone in the reverse order in which they were originally performed; for example, in Fig. 2.3, c_4 must be undone before c_3 can be undone. In non-linear undo, some commands can be skipped for undo as long as the current state of the model permits it; for example, c_3 could be undone without undoing c_4 (note that the resulting state would be different from both Q_2 and Q_3).

Since commands from each user are usually mixed in the command history list in multi-user applications, local undo requires a nonlinear type of undo. Nonlinear undo in turn is supported by the command pattern (storing the new state of a CAD feature in a command object) and the Memento pattern (storing the previous state of a CAD feature in the command object).

2.3.5 CAD Feature Dependencies

In CAD, features can have multiple parent features (features it references) and multiple child features (features that reference it; see Fig. 2.5). Fig. 2.6 shows a sketch (at the base of the model) that has multiple child and grandchild features. Fig. 2.7 shows an extrude that has multiple parent and grandparent features.

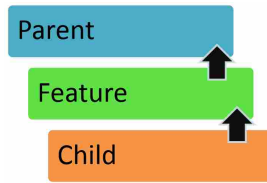


Figure 2.5: A feature references its parent feature, and its child feature references it.

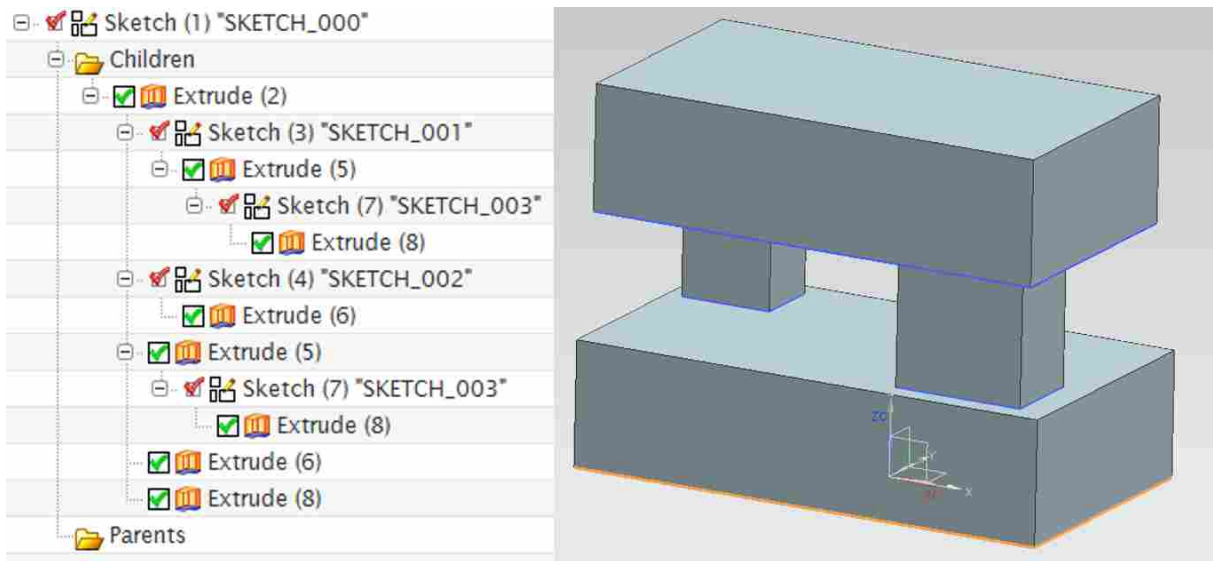


Figure 2.6: Child features of a sketch feature.

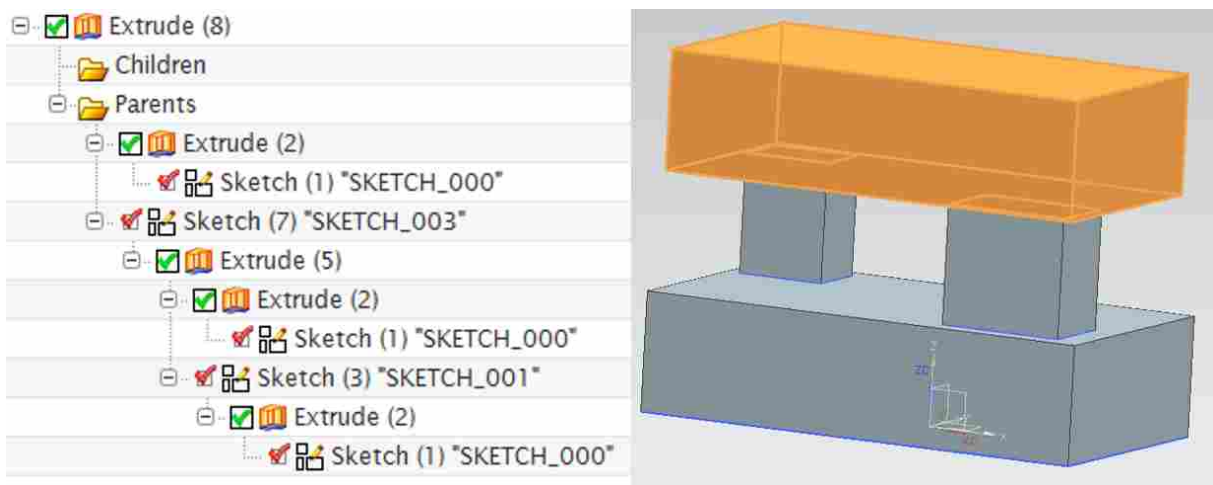


Figure 2.7: Parent features of an extrude feature.

2.3.6 Undo Conflicts

As in other multi-user systems, conflicts can occur during local undo in multi-user CAD due to its nonlinear nature and due to dependencies between commands from different users [64]. Due to the high number and complexity of dependencies between CAD features, these conflicts are more likely to occur and are more difficult to detect than conflicts in many other types of multi-user applications. A conflict occurs in local multi-user undo when:

- the undo does not or cannot provide the expected result due to commands that were performed by other users after the command that is being undone, or
- the undo affects the results of commands that were performed by other users after the command that is being undone (in this case undo can cause unintended consequences for either the local user or for other users).

Consider the following history list, where A_i refers to commands performed by User A, and B_i refers to commands performed by other users:

$$A_1 B_1 A_2 B_2 B_3$$

When User A performs an undo command, local undo will attempt to revert the A_2 command (in this notation, the undone command can be represented using a single quote, i.e. A_2'). If A_2 is independent of B_2 and B_3 , then A_2' will not produce a conflict. However, if there are dependencies between A_2 and either B_2 or B_3 , then A_2' may produce a conflict.

2.3.7 Syntactic and Semantic Conflicts

Both syntactic and semantic conflicts can occur during local undo in multi-user CAD. Contero et al. suggest that syntactic deficiencies in CAD are deficiencies in the mathematical validity of the model (e.g. invalid CAD features), and semantic deficiencies are aspects of the model that fall short of the user's design intent, despite being mathematically valid [26].

Example of a Syntactic Undo Conflict

Consider the following example of a syntactic undo conflict, referring to Figure 2.8: An extruded block exists with an edge blend around its top edges. User A deletes the edge blend,

leaving the extruded block in place (A_1). User B then deletes the extruded block (B_1). User A then attempts to undo his own previous command in an attempt to restore the edge blend (A_1'). This produces a syntactic conflict because the edge blend cannot exist in a syntactically valid state without the prerequisite existence of its parent feature, the extrude.

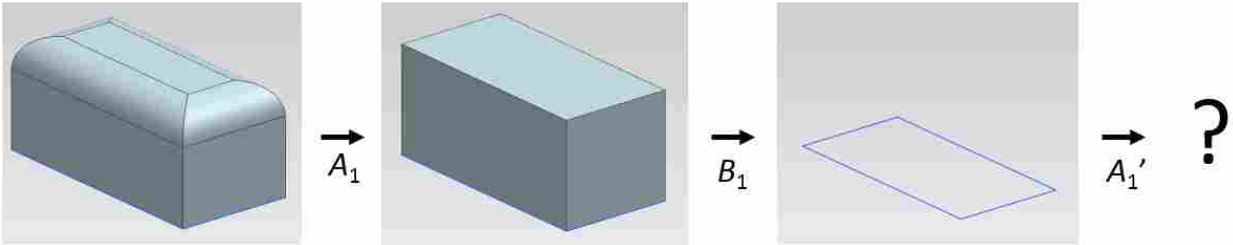


Figure 2.8: Example of a syntactic conflict.

Example of a Semantic Undo Conflict

Consider the following example of a semantic undo conflict, referring to Figure 2.9: An extruded block exists that is 1 unit long. User A extends the length of the block to 2 units (A_1). User B then creates a sketch on the right face of the first block (B_1) and extrudes that sketch to the right to create a new block (B_2) that is 2 units long. Moreover, suppose that User B made the second block 2 units long because his intention was that the two blocks have a combined length of 4 units. User A then undoes his own previous command to revert the first block’s length back to 1 unit (A_1'), resulting in a new combined length of 3 units. This produces a semantic conflict because the result conflicts with User B’s intention, which User A may not have been aware of.

2.3.8 Related Research

Undo/redo has been studied extensively for multi-user text editing [56–58,65–68]. In multi-user text editing, the document is essentially one long linked list in which the value of each element of the linked list is a character. So a document can essentially be represented as a long parent-child chain, since every character has one parent (the character before it), and one child (the character after it). Because every character is linked in a single chain, every comparison between two

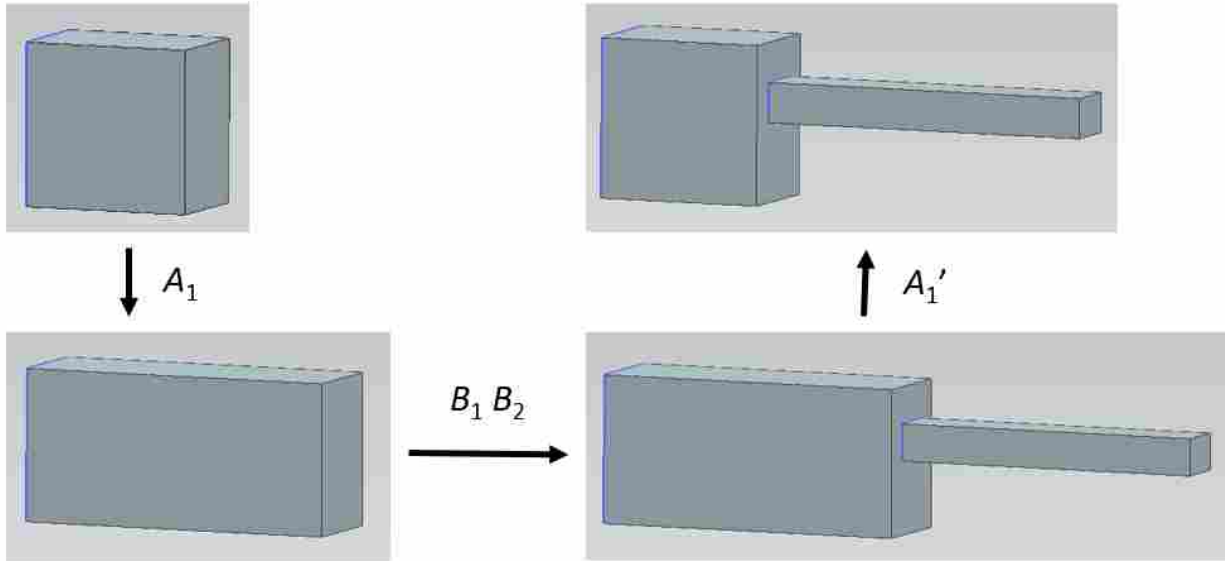


Figure 2.9: Example of a semantic conflict.

commands from different users will reveal a conflict, albeit a simple one. This conflict scenario can be automatically resolved using a method known as operational transform (shifting the insertion/deletion point of the command's character based on the number of characters inserted/removed by other commands). The only conflict scenarios not handled by operational transform in basic multi-user text editing are when two commands operate on the same position or adjacent positions [57]. Reasonably simple methods have been developed to handle these scenarios as well.

Conflict detection is different and more difficult in multi-user CAD than in multi-user text editing, because each CAD feature can and frequently does have multiple parent features and multiple child features. Some research has been conducted for detecting conflicts in multi-user CAD.

Liu et al. compare dependence between two concurrent commands based on whether the commands create Boolean operations that intersect one another or not [69]. This requires a CAD kernel-level comparison between the results of the two commands. When checking dependencies between the command being undone and more recent commands from other users, this would require executing the undo command and comparing the results of that command with every other command that has been executed by other users since the command that is being undone. It would also not catch all conflicts such as parametric dependencies between features that do not directly

involve a Boolean operation; for example, when a sketch is created on a face, the sketch is dependent on the face even though no Boolean operation connects the sketch and the face.

Cheng et al. compare the dependence between the command being undone and recent commands from other users based on whether the recent command from another user is dependent on (i.e. a “child” of) the feature being undone [45, 70, 71]. This method will detect some types of conflicts but fails to detect if prerequisite “parent” features still exist before allowing a feature to be undone to a non-deleted state. Cheng et al.’s method also requires the undo command to be checked for dependence with every more recent command from other users. Gao et al. use the same method as Cheng et al., but extend it apply to commands that involve multiple operations [64]. The same limitations apply to Gao et al.’s method as apply to Cheng et al.’s.

2.3.9 Summary

Although some research has been performed on undo conflicts in other types of software, very little previous research has been performed on undo conflicts in multi-user CAD, which is a much more complex challenge to solve.

2.4 Administering Data Access in Multi-User CAD

Data access control is an important part of collaboration between individuals and companies in engineering industry since CAD designs can, and often do, contain critical proprietary data: “The engineering data created using [3D CAD] is often among a company’s most valuable intellectual property,” said Joyce Durst, President and CEO of Pinion Software. “When this information is shared without continuous protection as part of design collaboration, outsourced manufacturing, or other technical business processes, it could become subject to unintended or illegitimate use that could have significant consequences to your business;”. “With the growing trend to distribute CAD files to customers and suppliers, a company not only has to be concerned about the safety of their intellectual property in the hands of their own employees, but also in the hands of all those other companies’ employees,” [37].

Many engineering companies are particularly sensitive to the security of the intellectual property contained in their CAD data, since much of their data is proprietary, and since some

data are subject to national regulations such as Export Administration Regulations (EAR) [38] and The International Traffic in Arms Regulations (ITAR) [39]. Also, as engineering designs become more complex, engineering work is increasingly being outsourced between companies and even countries.

Multi-user CAD applications need to incorporate methods to protect design data to prevent loss of valuable proprietary data. Since by nature a multi-user CAD program exercises a high degree of control over the CAD system, it should be able to restrict certain user actions while retaining the support for multi-user collaboration in a full-featured CAD model.

For example, if a large U.S. OEM needed to outsource a portion of a certain design to a supplier in Europe using multi-user CAD, they would want a method to allow the foreign supplier to access only the necessary portions of the design. The supplier should only be able to view and edit what is necessary to design their part of the model, while restricting viewing/editing access to all other portions of the model. Integrating methods like these into a multi-user CAD system would greatly facilitate collaboration and security in multi-user design across a supply chain.

2.4.1 Access Control Models

Three dominant access control models are Role Based Access Control (RBAC), Discretionary Access Control (DAC), and Mandatory Access Control (MAC) [72]. Wikipedia describes these three access control models quite simply [73]: “RBAC allows access based on the job title. For example, a human resources specialist should not have permissions to create network accounts; this should be a role reserved for network administrators.” “In DAC, the data owner determines who can access specific resources. For example, a system administrator may create a hierarchy of files to be accessed based on certain permissions.” And, “In MAC, users do not have much freedom to determine who has access to their files. For example, security clearance of users and classification of data (as confidential, secret or top secret) are used as security labels to define the level of trust.”

Hybrids or layering of these policies are also possible. Access control can also support a hierarchical inheritance structure wherein one attribute can inherit the access rights of another attribute: for example, a Senior Engineer could inherit all of the access rights of a Junior Engineer.

2.4.2 Related Research

In an attempt to secure company and product data, original equipment manufacturers (OEMs) sometimes send their suppliers lightweight versions of CAD part or assembly files: “Different views of the information can be produced for specific users and combined with different lightweight representations according to the corresponding level of security and information requirements of the target user,” [74].

Product data management (PDM) systems are the current standard for managing which individuals, teams, or organizations can access which CAD models, including who has read/write access [37,75]. In multi-user CAD, CAD files are no longer managed by a traditional PDM system; rather, they are stored in a database and managed by the multi-user CAD client and server. In multi-user CAD, individuals from multiple companies may want to collaborate together on the same model while restricting some of the viewing or editing permissions of users that do not own the data. Since a multi-user environment has a high degree of control over the CAD system, it should be able to restrict certain user actions while retaining support for multi-user collaboration in a full-featured model.

Teng et al. studied a hybrid MAC and RBAC access model for multi-user CAD part files [40]. They also studied encrypted transport and integration of existing directory services (the LDAP protocol and Microsoft Active Directory domain services) to support their access model. Teng validated his system in the BYU NXConnect prototype using parts and assemblies where various components were configured with different access privileges. Teng did not study the DAC access model. Google Drive uses a DAC access model in which users who own a file can assign permissions to other users to access that file [76].

Role-based permissions in multi-user CAD based on reducing level-of-detail (LOD) by hiding feature detail has been studied by Cera et al. [77–82] and Chu et al. [83]. In this method, “design features are selectively hidden at each level from certain participants, depending on their actual needs and individual accessibility in the collaboration,” [83]. Identified in these studies were multiple resolution techniques for shape preservation and data transfer, and “most mesh-based methods were focused on graphics rendering with less emphasis on design intents,” [83].

In multi-user CAD these level-of-detail reduction methods could support collaborative viewing of the model if the level-of-detail could be automatically reduced without manual in-

put; otherwise the other users would see the new features as other users created them before the level of detail was reduced. And if users were not shown the full level-of-detail, it would be difficult and in some cases impossible for them to collaboratively edit the simplified portions of the model. Thus Cera’s research supports collaborative viewing of the model, but requires automatic feature simplification methods. It does not support collaborative editing of simplified portions of the model.

Moncur et al. have studied methods for restricting permissions to edit and view features [16,41]. Red and Marshall et al. have studied methods for restricting permission to edit spatial regions of a multi-user CAD model using geometric constraints [42,43]. In Red et al.’s method, “the user’s feature selection cursor is only allowed to select features that lie within the assigned and constrained user region,” [42].

Additional opportunities exist for facilitating secure collaboration in multi-user CAD. Three important restriction types that will be important to support in multi-user CAD are the capabilities to (1) manage, (2) edit, and (3) view model data. A model can be segmented in various ways for applying these restrictions; for example, a model can be segmented by spatial region, by part, or by feature. Table 2.2 shows which researchers have studied which types of data access control mechanisms (“French” indicates my contributions).

Table 2.2: Multi-user CAD permission types and researchers who have studied each type.

Permissions Type	Manage	Contribute	View
Level-of-detail reduction	Cera, Chu	Cera, Chu	Cera, Chu
Spatial region	<i>French</i>	Red/Marshall	<i>French</i>
Part	Teng (RBAC, MAC) <i>French (DAC)</i>	Teng (RBAC, MAC) <i>French (DAC)</i>	Teng (RBAC, MAC) <i>French (DAC)</i>
Feature	Moncur	Moncur	Moncur

Specifically, the following methods have not yet been developed:

- Administer spatial permissions in a multi-user CAD model.
- Restrict users from viewing a spatial region in a multi-user CAD model.

- Administer permissions using a Discretionary Access Control (DAC) model, including restricting users from contributing to (editing) or viewing a multi-user CAD model.

2.4.3 Summary

Additional methods need to be studied and developed to administer data access in multi-user CAD. These methods will be essential to the practical use of multi-user CAD in engineering industry due to the intense security practices and regulations that affect this industry. They will allow organizations to reduce the amount of proprietary data in a multi-user CAD environment that is exposed to risk of theft or accidental disclosure. It will address a major collaboration challenge since it will allow for the sharing of partial design data without revealing proprietary data; otherwise, less design data might be allowed to be shared due to security policies and regulations.

CHAPTER 3. OBJECTIVES

Four important research efforts need to be studied before multi-user CAD can be used effectively. Two of these research efforts will understand the human side of multi-user design collaboration, and two of them will address the technical side of multi-user design collaboration.

The research objectives that will understand the human side of multi-user design collaboration are the following:

- Better understand collaboration challenges in engineering industry to understand how multi-user CAD will help address some of those challenges.
- Understand how a multi-user environment affects design collaboration.

The research objectives that will address some important technical challenges in multi-user design collaboration are the following:

- Develop a classification of types of undo conflicts in multi-user CAD and develop a method for detecting those conflicts.
- Develop additional methods for administering and restricting data access in multi-user CAD.

Figure 3.1 shows the two human collaboration challenges and two technical collaboration challenges that this dissertation addresses.

These four studies of human and technical collaboration challenges will constitute a balanced effort for facilitating the effective use of multi-user CAD in a modern engineering environment.

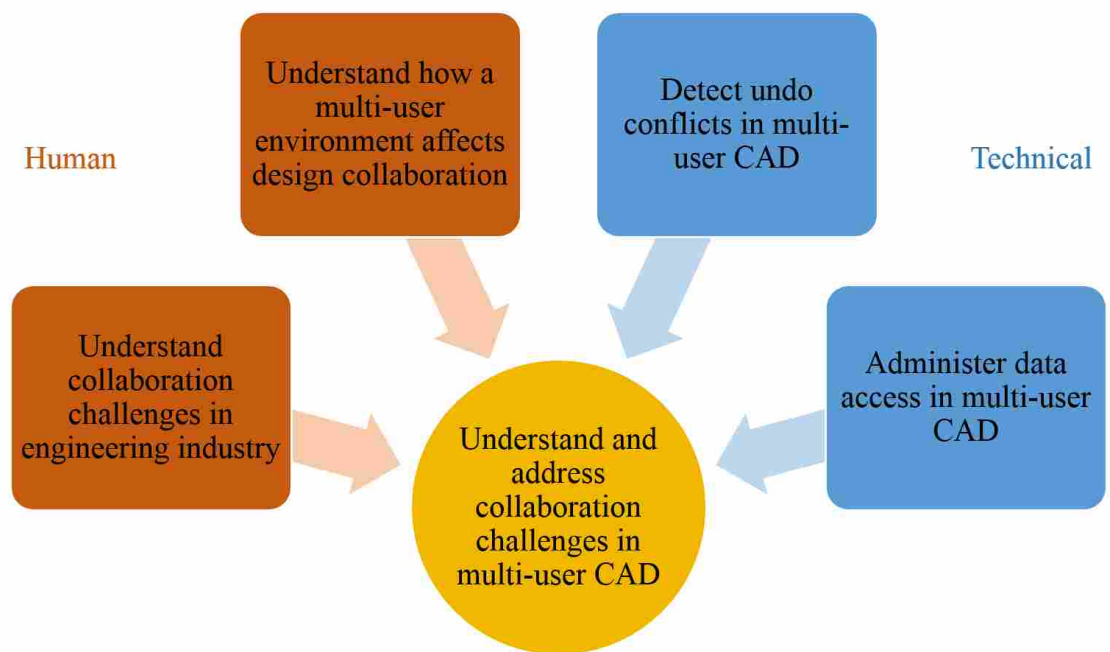


Figure 3.1: Some important human and technical design collaboration challenges in multi-user CAD (Human collaboration challenges are marked orange; technical collaboration challenges are marked blue).

CHAPTER 4. A STUDY OF DESIGN COLLABORATION CHALLENGES IN ENGINEERING INDUSTRY

4.1 Method

My fellow researchers and I conducted a study in 2011 to understand modern design collaboration practices and challenges in engineering industry, how industry copes with those challenges, and how multi-user CAD will help address some of those challenges. We also wanted to better understand the potential role of multi-user CAD in engineering industry and identify potential strategies for using multi-user CAD effectively in industry.

Data was gathered by visiting companies to survey, interview, and observe engineers and their collaboration. We asked engineers questions about the collaboration challenges they face, their team dynamics, team organization, communication and technical tools, and various other topics related to collaboration. This study is called the “E-Handbook of Collaborative Engineering,” and is referred to hereafter as the “E-Handbook study.”

4.2 Results

A total of 9 companies and 144 engineers participated from various engineering industries including aerospace, defense, energy, manufacturing, and medical. Of these, 5 companies participated in interviews and on-site visits and observations; 15-30 engineers were interviewed at each of these companies. 4 companies participated only in surveys; 6-9 engineers were surveyed at each of these companies. Following the interviews, we compared our data for each interview question by company. The company names have been kept anonymous.

4.2.1 Demographics

86% of participants were male; 14% were female. 57% of study participants were team leaders. Participants' ages ranged from 23 to 66 years old, with a median age of 42. The median length of time participants had worked at their current company was 11 years. The median number of years of professional experience was 19 years. English was by far the dominant language among study participants: All participants used English at work, and only 3% of respondents also used a language other than English at work. The majority (53%) of participants had a four-year degree, and another 31% of respondents had either a master's or doctorate degree (see Fig. 4.1).

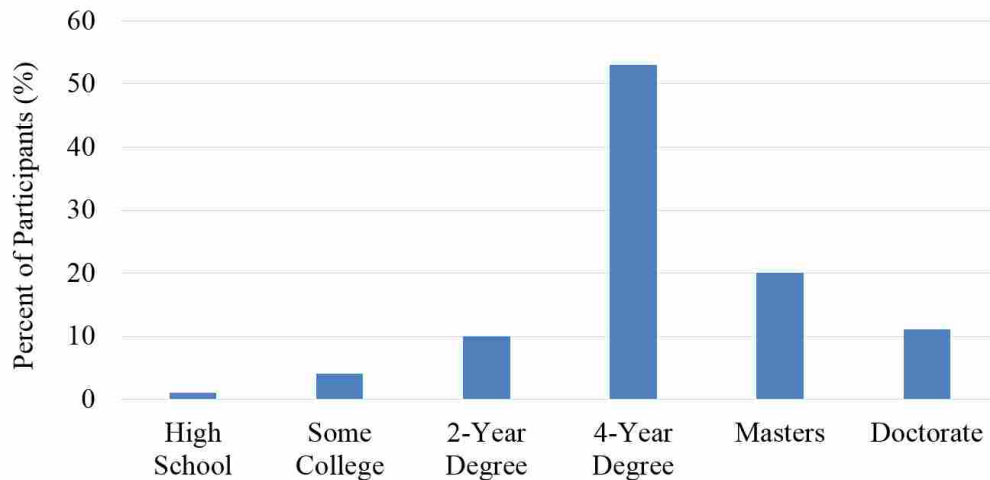


Figure 4.1: Education levels of study participants.

4.2.2 Team Size

The median team size in the E-Handbook study was 9 people, typically ranging between 4 and 10 people 4.2. About 12% of participants said that their team had more than 30 people. Research indicates that the best team size is somewhere between 3 and 7 members, although this is typically given as the size of the smallest sub-team within a group, since larger teams can be difficult to manage [84, 85].

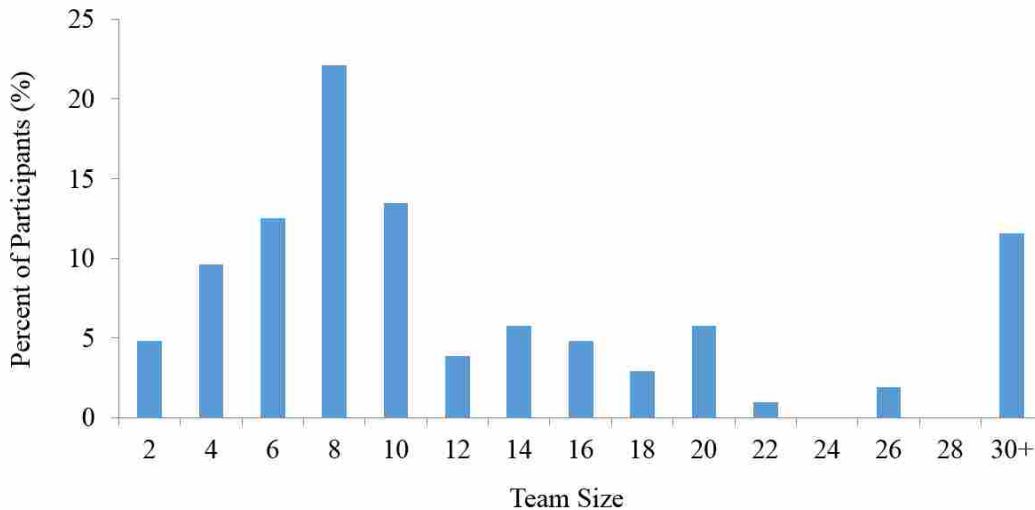


Figure 4.2: Histogram of team sizes (defined as “those you work directly with”).

4.2.3 Time Spent Collaborating

We asked interviewees what percent of their time is typically spent collaborating (defining “collaborating” as working with others as opposed to working individually). The results are shown in Figure 4.3. The median reported time spent working with others was 60%, and there are many people who claim to spend between 70 and 90% of their time collaborating. This demonstrates that engineers spend a very large proportion of their collaborating and working together; therefore, if collaboration can be streamlined (for example, by using multi-user CAD) then an increase in time savings and design quality may be realized.

An interesting characteristic of the team size histogram (Fig. 4.3) is that the distribution appears to be bi-modal, with one mode at 55% and a second mode at 70%. We tried separating out the team leads and technicians, which appeared to at least partly explain the two modes, as can be seen in Figure 4.4: Team leads appear to spend about 15% more time collaborating than technicians. This makes sense since team leads often act as a communication hub for the team. Team leads likely also attend more meetings than technicians and also spend time reporting to higher-level management. So streamlining collaboration (for example, by using multi-user CAD) will hopefully ease the communication burden on team leads so they can focus on other things.

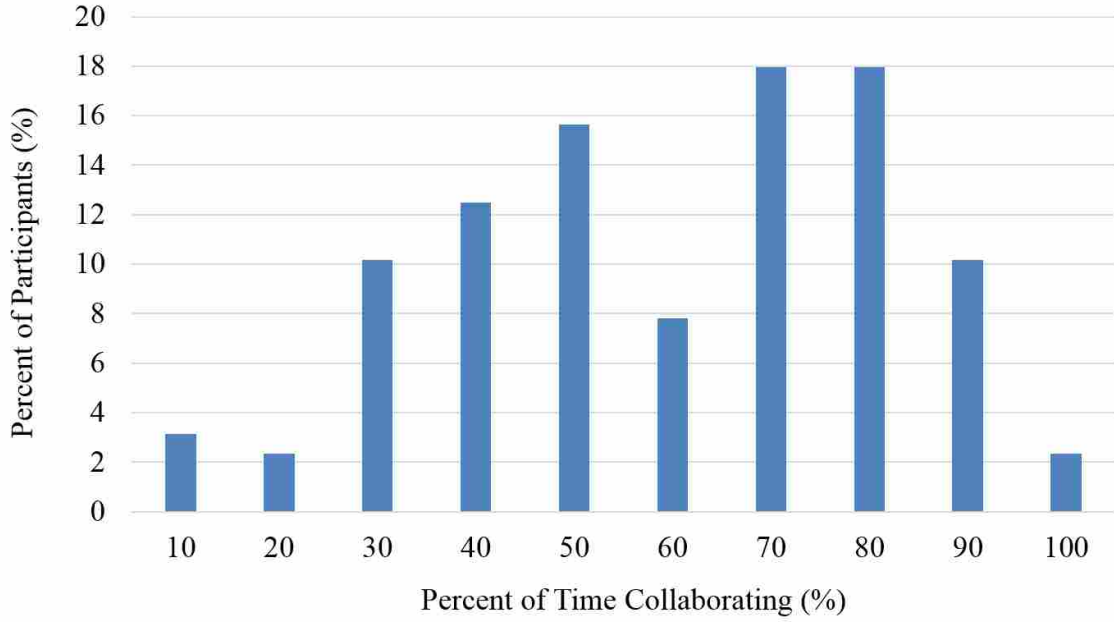


Figure 4.3: Percent of time spent collaborating (working with others).

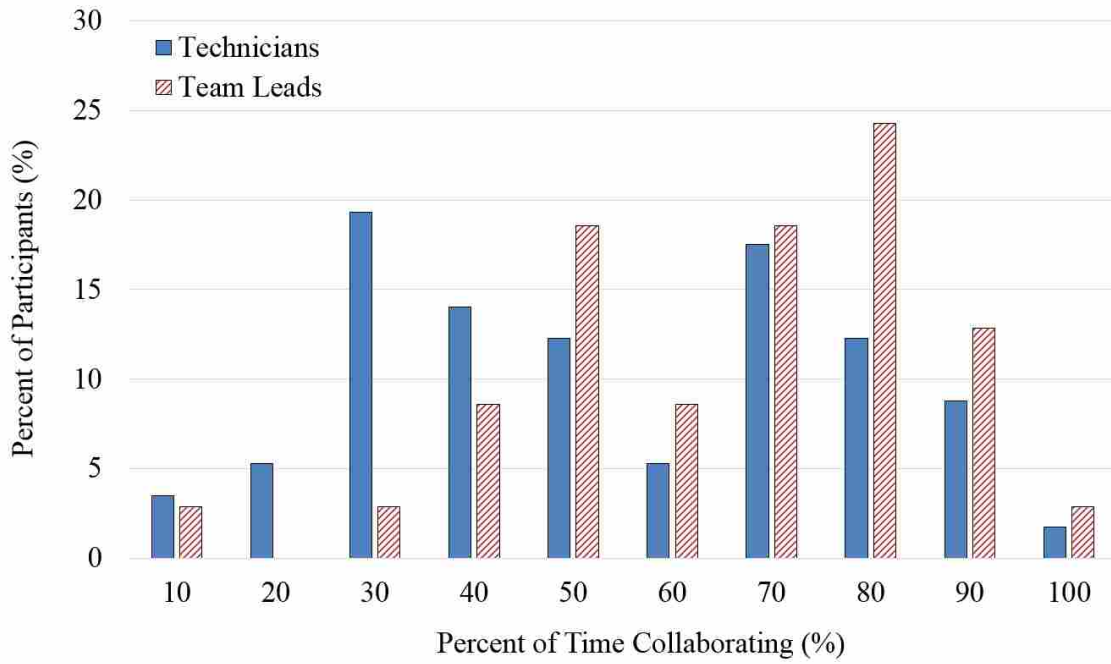


Figure 4.4: Percent of time spent collaborating by role.

4.2.4 Effectiveness of Team Collaboration

We asked interviewees how they rated their team’s ability to collaborate on a scale from 1 to 10: See Figure 4.5. 65% of participants rated their team at a 7 or 8. This question is fairly subjective, but it shows that most participants judged their collaboration to be quite effective overall, but that they also think there is room to improve. One participant said the following about the need for him and his team to improve their collaboration:

“We aren’t as collaborative as we should or need to be... We do well in our discipline but we aren’t focused from the perspective of a teammate... It imposes upon the leader an additional responsibility to make the team function to make sure the right person does the right work and the right data is shared with each other.”

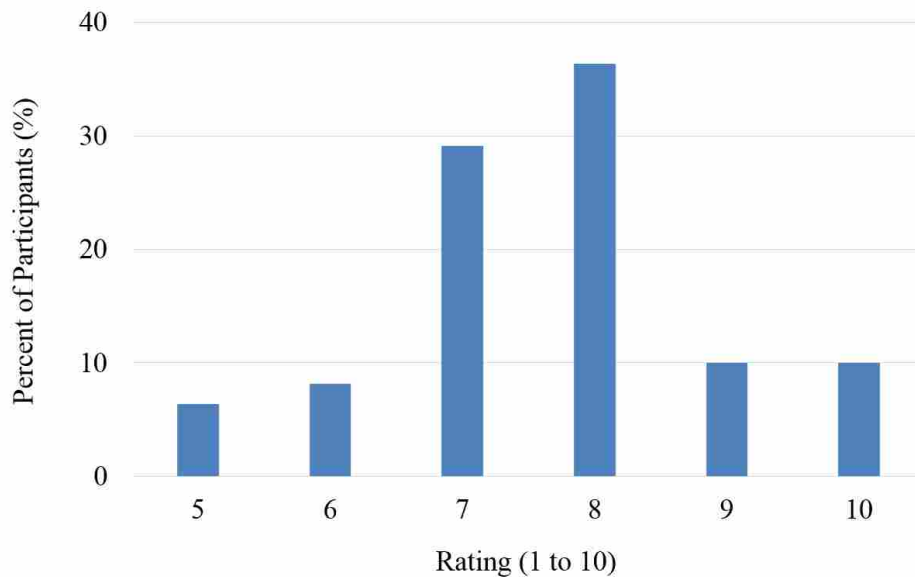


Figure 4.5: Team collaboration effectiveness ratings (scale from 1 to 10).

We also asked participants what was effective or ineffective about their team’s collaboration and categorized their responses. Figures 4.6 and 4.7 show the participants’ responses. It is clear that overall the participants thought that communication was the biggest factor in making their team effective or ineffective.

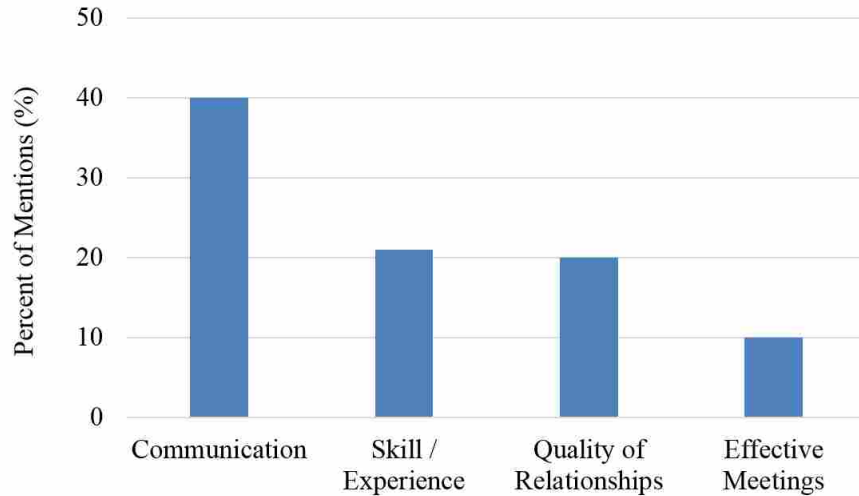


Figure 4.6: Responses to the question, “What makes your team collaboration effective?”

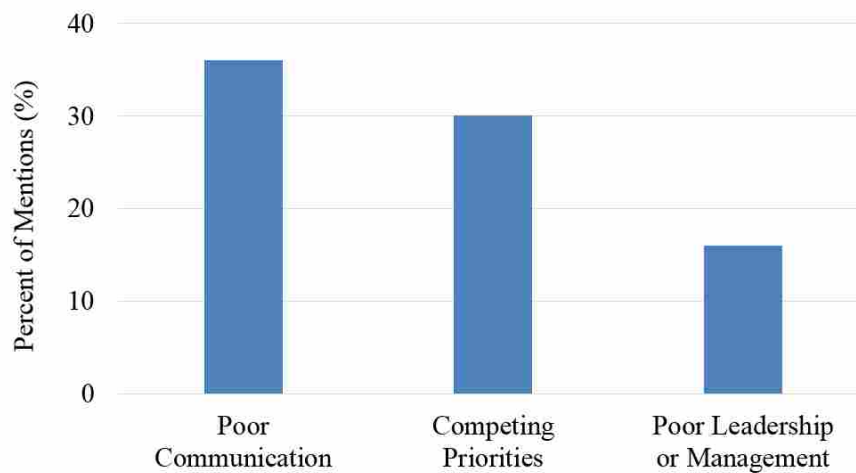


Figure 4.7: Responses to the question, “What makes your team collaboration ineffective?”

One participant gave some examples of communication errors that can occur: “Important messages are sometimes lost. The workers do not always get everything that is needed to perform optimally on a project. Similarly, the project manager does not always get all the information needed to optimally perform.” Another participant highlighted the importance of receiving information first hand: “Communication is key. I harp on that... You need first hand communication, not second hand. Interpretations get messed up.” And another participant gave yet another example of what how communication tools can be used inappropriately:

“We do a lot of emailing, and cell phones work as part of normal business. They can be good tools and augment face to face communication. But it can be problematic if people rely on those as the primary forms of communication. They should be used for the tools they are as opposed to a crutch.”

We propose that multi-user CAD will help teams communicate better by providing them with a shared 3D context in which to have design discussions and see the latest 3D design information first hand in real-time, and by lowering the communication overhead of passing around CAD files and making sure they are using the correct versions of those files. However, teams will need to make sure that they use collaborative engineering tools appropriately and not as a crutch to other forms of communication, including face-to-face interactions.

In addition to “poor communication”, participants also heavily mentioned “competing priorities” and “poor leadership or management” as making their team collaboration ineffective. These issues could perhaps be lessened by multi-user CAD—or at least by multi-disciplinary multi-user CAx—because if an entire team (including all disciplines) were working on the same shared set of data, then team members could instantly see changes that other disciplines or engineers made and thus be aware of how those changes might affect their own work. It would also provide a shared context in which they could discuss their competing priorities and make changes together in real-time.

Ferrazzi encourages virtual teammates to use technology to connect and build relationships [86] which can help reduce the burden on the team leader. The problem of poor management or leadership may also be lessened by having a shared multi-user environment, since it might reduce the team’s reliance on the leader’s role as a coordinator and communication hub, and instead allow the team to be more self-lead. This goes right to the heart of the comment mentioned earlier in this chapter, repeated again for convenience:

“We aren’t as collaborative as we should or need to be... We do well in our discipline but we aren’t focused from the perspective of a teammate... It imposes upon the leader an additional responsibility to make the team function to make sure the right person does the right work and the right data is shared with each other.”

When a team leader is freed from some of his responsibilities as the collaboration hub of the team, then poor management and leadership may become less of an issue. Note, however, that team leaders can also help improve collaboration effectiveness: Some team leaders reported assigning mentors, training on communication tools, and setting standards for how to organize or transfer files between team members in an effort to improve collaboration on their team.

Barriers to Collaboration

We also asked participants, “What are the greatest barriers to collaboration in your work, overall?” We categorized their responses which are displayed in Figure 4.8. Overall, “teaming / team dynamic issues” was the most commonly mentioned barrier to collaboration, followed by “schedule conflicts” and “work load / resources.” While multi-user CAD will not be a panacea to all teaming problems, it may help somewhat by reducing the collaborative friction and frustration that is caused by not having a common model to work in and by not having an opportunity to catch each others’ design mistakes until later in the design process.

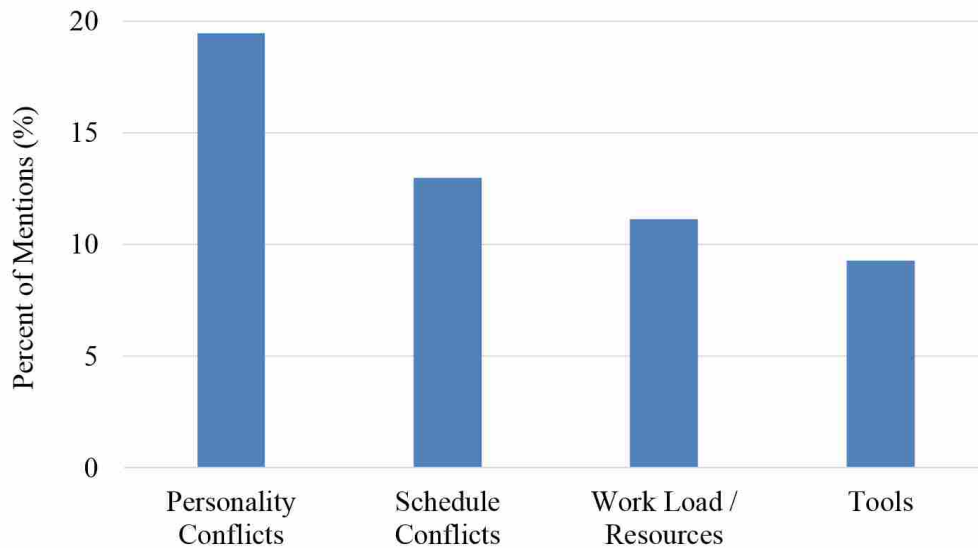


Figure 4.8: Barriers to collaboration.

“Teaming / team dynamic issues” mostly refers to personality clashes and team conflicts. Some participants also reported experiencing role confusion and unwillingness to give or receive help.

In regards to schedule conflicts, most if not all engineering companies use a calendar tool (e.g. Outlook calendar) to help facilitate scheduling, but since engineers spending on average 60% of their time working with others (see section 4.2.3 above), a lot people’s time is in demand by multiple other people or groups. One respondent showed us his weekly calendar, and there were some times in his week when there were three or even four overlapping meetings that he was “supposed” to attend. Other study participants confirmed that this is not uncommon.

Workload/resources refers to the fact that some interviewees felt like they personally had so much to do that they had to focus on their personal tasks rather than collaborate with others. A multi-user design environment could help the team and any supervisors be more aware of each others’ activities so they could focus on the team’s priorities as a whole instead of zooming in on their own personal tasks.

“Tools” was the fourth most-frequently mentioned barrier to collaboration. Multi-user CAD and other related collaboration and design tools could assist greatly in removing this perceived barrier.

4.2.5 Meetings

Meetings are an important aspect of team collaboration in today’s engineering environment. Formal meetings, ad-hoc meetings, design reviews, and meetings with external parties (e.g. customers, suppliers) are held on a regular basis and can be a more or less effective use of people’s time.

Time in Meetings

The median amount of time respondents claimed spending in formal meetings was 30%, with a fairly wide distribution of responses: See Figure 4.9.

We also filtered participants’ percent of time spent in meetings by job role, and found that technicians usually spend about 20% of their time in formal meetings, while team leads and

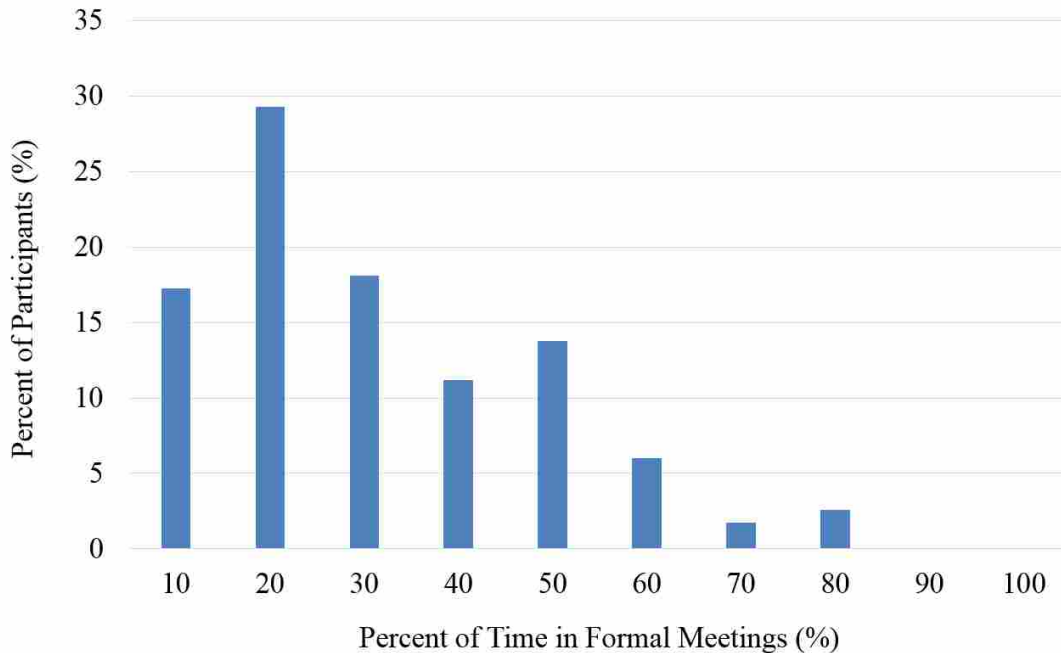


Figure 4.9: Percent of time spent in formal meetings.

corporate-level engineers spend anywhere from 20% to 50% of their time in meetings, with a fairly even distribution (see Fig. 4.10).

Effectiveness of Meetings

In general, ad-hoc meetings were seen as being very effective, whereas formal meetings were only seen as being somewhat effective. 60% of participants felt like they spend too much time in formal meetings.

Table 4.1 summarizes the differences between formal and ad-hoc meetings. Participants reported that formal meetings can be more successful when the right people are invited to the meeting, an agenda is prepared and followed, and discussion is kept on-topic. Ad-hoc meetings were typically reported to be very effective because the whole point of the meeting is to address a specific question or need, and only the necessary people are included. Ad-hoc meetings were reported to last an average of 20 minutes, whereas formal meetings last an average of 60 minutes.

interviewees told us that having a prepared agenda, keeping the meeting on topic, and only inviting the necessary people to the meeting were keys to holding an effective meeting. Thus it

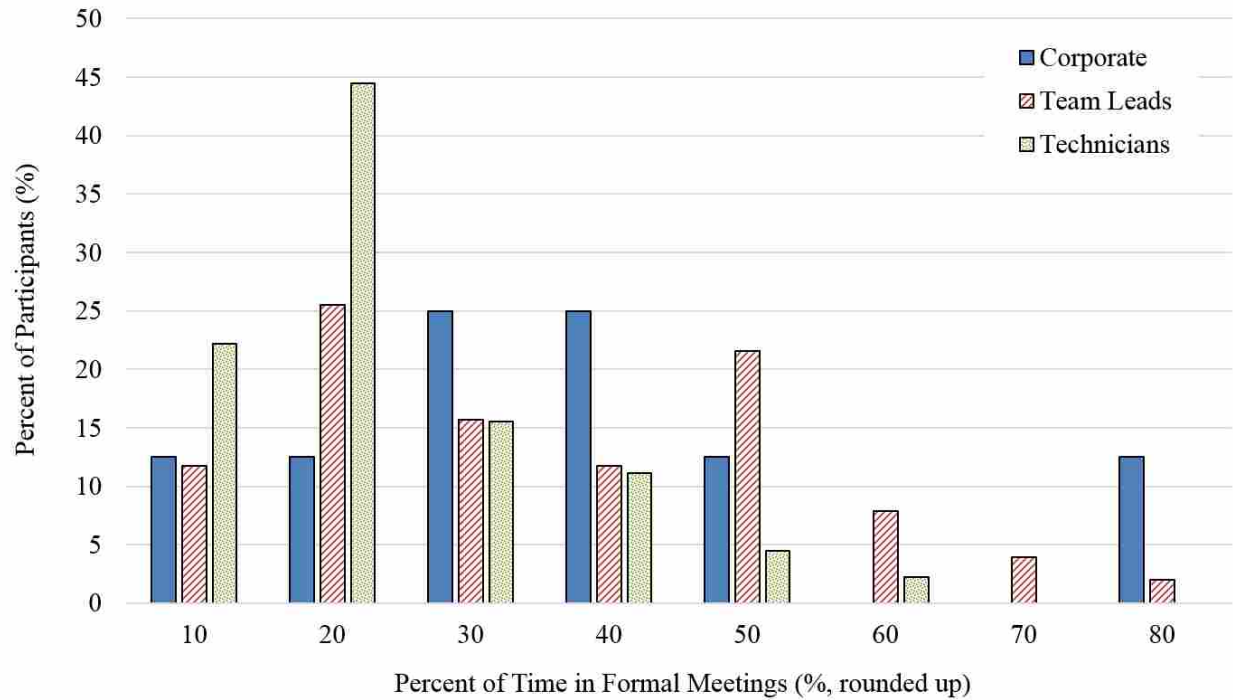


Figure 4.10: Percent of time spent in meetings by role.

Table 4.1: Effectiveness and typical length of formal and ad-hoc meetings.

	Formal Meetings	Ad-hoc Meetings
Average Length	60 min	20 min
Effectiveness	Somewhat effective	Very effective
What is effective	Agenda, focus, goals	One-on-one, relevance, openness
What is ineffective	Off-topic, unnecessary information, too many attendees	Social loafing

is likely that it is the quality of the meeting or getting the right set of attendees that determines peoples' satisfaction with their meeting time.

Meeting Room Availability

Low meeting room availability was a commonly mentioned challenge at all companies. Part of the problem was having enough meeting rooms, and part of the problem was scheduling

the meeting rooms. One company had a large open atrium with lots of tables for the purpose of accommodating ad-hoc discussions. This room was highly spoken of by study participants.

4.2.6 Virtual Teaming Challenges

We only had the opportunity to interview a couple of engineers that regularly worked from home or away from the rest of their team, but a many participants said they regularly work with some remote teammates. The median reported number of remote teammates that study participants worked with was 5. Teammates located on the same campus but in a different building were reported as being, for all intents and purposes, virtual teammates. This concept is confirmed by Ferrazzi: “Even colleagues on different floors in the same building might be considered physically distant,” [86].

Interviewees revealed several challenges that are commonly encountered when with working with a virtual team or teammate (each of these challenges is discussed in further detail below):

- Language differences
- Time zone differences
- Security
- Tool incompatibility
- Trust and social relationships

Language Differences

We only interviewed teams working in the United States, but many of these engineers reported working collaboratively with teams from Asia, Europe, or other areas of the world. Language barriers can include more than direct translation problems: Some local expressions can be lost in translation. One example of this is that a U.S. engineer asked a non-U.S. engineer if he understood some instructions that were given. The non-U.S. engineer replied, “Yes.” The U.S. engineer thought this meant that the other engineer understood the instructions, but the other engineer really meant that he was paying attention—not that he understood! To help fix this type of

miscommunication, the study participant recommended asking the other person to repeat back the instructions given to him or her.

Some participants also reported that email can sometimes be a more effective communication medium than telephone, since more time can be taken, if necessary, to try to interpret and understand an email; whereas, on the phone even a strong accent in the listener's native language can inhibit clear communication. Multi-user CAD may facilitate language barriers somewhat by providing a shared context for a discussion; however, language differences will still be difficult to deal with using synchronous communication tools such as VoIP and chat, which are the communication tools that most naturally complement multi-user CAD.

Time Zone Differences

Time zone differences can make it difficult to have meetings over the phone, since one team has to stay late or arrive early to work to accommodate the time zone of the other team. When this is necessary on a regular basis, some interviewees mentioned that it is courteous to alternate which team has to adjust their schedule to make the meeting so the same team isn't always being inconvenienced. Malhotra et al. confirms the benefit of this strategy: "Ensuring that everyone 'suffered' equally from working in a geographically distributed world also created trust. The team leaders rotated the times at which weekly audio-conferences were held so that everyone (at some point in the teams life cycle) would experience the pain of a late night or early morning meeting," [87].

If a project utilizes time zones as a way to support multiple work shifts, then multi-user CAD may make it a little easier to hand off work between teams since its architecture naturally supports a rich revision history that could be viewed by the team on the next shift, so they can more easily see the changes that the previous shift made.

In addition, virtual teammates don't have the same opportunities for ad-hoc socialization and unplanned meetings: One participant said, "The communication tools are not anywhere near as effective as being close to folks. When you are close to people there is a lot of unplanned communication, which is good. When people are 8 time zones away you have to call or email them. And if they are swamped with emails that doesn't help."

Security

Security is also a major concern for virtual teams, especially when dealing with outsourcing teams that are from a different country. Export control laws and rules such as the U.S. Export Administration Regulations (EAR) and The International Traffic in Arms Regulations (ITAR) limit the amount of information that may be shared with an external team [38,39]. And even information that can be shared may need to go through an approval process which can take time.

Several participants mentioned that security was an issue in dealing with foreign teammates. Sometimes the person designated to approve some information for sharing with a foreign team is stuck in meetings or otherwise unavailable for hours or even days, thus delaying the communication of information with the foreign team. Participants also reported having to use slow transmission methods for sending sensitive information to foreign entities: for example, using a secure courier to send a CD or external hard drive instead of emailing or otherwise transmitting a sensitive file. This was especially the case for large files.

Since multi-user CAD naturally enables virtual teaming, it will be especially important for it to provide a way to appropriately protect sensitive data. This will be especially important for virtual teams with members in different companies and countries.

Tool Incompatibility

Tool compatibility is another significant challenge that interviewees reported in virtual team collaboration. When different tools are used for the same purpose, data needs to be translated which often reduces the quality of the data. Ferrazzi says, “In our experience, even top-notch virtual teams—those with the most-talented workers, the finest leadership, and frequent touchpoints [in-person meetings]—can be felled by poor technology” [86]. The Research Triangle Institute prepared a report for the National Institute of Standards and Technology (NIST) in 1999 which claimed that, “imperfect interoperability imposes at least \$1 billion per year on the members of the U.S. automotive supply chain,” [88]. While multi-user CAD itself will not necessarily fix this problem, the additional collaborative opportunities it affords may increase the incentive for companies to try to use the same multi-user tools.

Trust and Social Relationships

Levi says, “Developing trust among team members is the foundation for both task and social communication in virtual teams,” and “Virtual teams start with lower levels of cohesion and trust among members... Technology constraints on communication create confusion about taskwork and limit development of social relations,” [89]. Milstein, a principal at the consulting firm Argea, said in reference to virtual teamwork, “The development of trust is quite often the single most important tool in overcoming barriers and obstacles,” [90]. Ross suggests several methods for increasing and maintaining trust in virtual teams, including holding face-to-face meetings, setting clear goals, increasing work transparency, and providing feedback [90].

One participant in the E-Handbook study commented, “I think that camaraderie—the fact that we all have a set purpose, a set goal that we all have to meet no matter how we get there—is what brings everyone together to make everything better.” Stanford business professor Margaret Neale confirms this: “The team can handle divergence without it eroding trust if everyone has the same goal,” [90].

Figure 4.11 shows the difference in participants’ familiarity with their local versus their remote collaborators.

The challenge of trust in virtual teams was mentioned a few times in interviews. A couple of study participants mentioned that when they start a new project in collaboration with a virtual team, they have at least one face-to-face meeting with their remote teammates at the start of the project so they can get to know one another better and establish more trust and understanding with each other. This is consistent with recommendations from organizational behavior research: Rocco showed that a pre-project face-to-face meeting can promote trust in virtual teaming situations since it promotes socialization [91]. Since multi-user CAD will facilitate and perhaps increase the use of virtual teaming, it is important that companies and teams continue to take steps to build social cohesion and trust within virtual teams.

4.2.7 Organizational Structure

Several types of organizational structures are reported in the literature; notable types include functional, divisional, and matrix [92,93]. Based on participants’ responses it appeared that

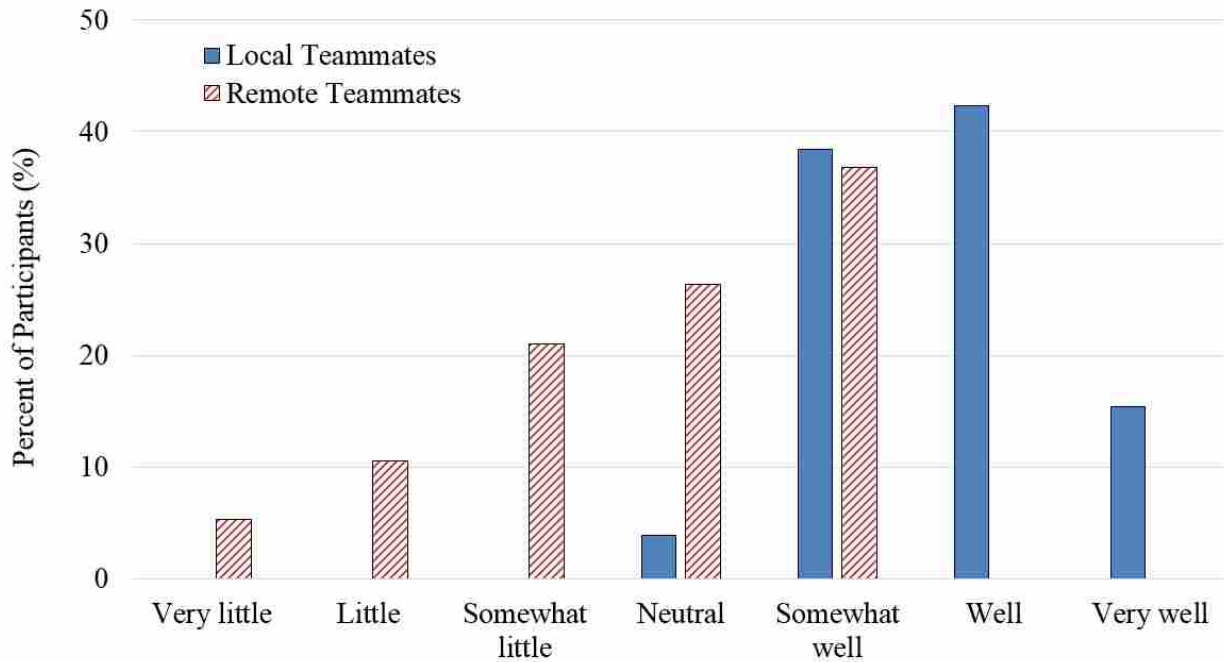


Figure 4.11: Familiarity with local vs. virtual collaborators (“How well do you know the people you work with?”).

most of the companies we visited used a matrix organizational structure. A matrix structure means that each employee has two supervisors: one functional and one project/product supervisor. For example, a structural engineer would report to both his project lead and to the head of structural engineering. The matrix structure seemed to help large engineering companies working on several complex projects to utilize individuals’ talents effectively both across the company and within a certain project.

Although the matrix structure is a good way to utilize talent effectively, it does take a toll on collaboration effectiveness: One respondent commented that “The matrix structure makes it somewhat difficult to integrate the plans and schedules of the various teams.” Other respondents mentioned the difficulty of balancing priorities since they were accountable to two different teams (i.e. their functional team and their project team). As noted earlier in this chapter, the second most frequently mentioned cause of team collaboration ineffectiveness was “competing priorities” (see Fig. 4.7), and it appears that this may be due in large measure to the dual accountability the matrix

organizational structure imposes. Being on two different teams also makes it even more difficult to colocate engineers with all of their teammates.

The more connected and collaborative the engineering and communication tools can become, the easier it will be for these functional and project teams and team members to collaborate and coordinate their competing schedules and priorities. Specifically, there appeared to be a need for a more effective calendar tool to make it easier to coordinate not only people's availability, but also the availability of conference rooms.

4.2.8 Communication Tools

Communication tools are extremely important to engineering team collaboration. Li et al. define two types of communication tools: "Synchronous tools enable real-time communication for users to work together in the 'same time-different place'," and "Asynchronous tools can enable communication and collaboration to happen in the 'different time-different place'," [94]. Both synchronous and asynchronous collaboration technologies are important in engineering collaboration, because sometimes people want to work on something at the same time, and sometimes they want to work on something at different times.

In Figure 4.12 we categorize several common communication tools as either synchronous or asynchronous. Although we did not identify multi-user applications as a type of "communication tool" in this figure, it is worth noting that they are also sometimes referred to as "synchronous collaboration" technologies [94–96].

Some of these communication tools may be useful to integrate with multi-user CAD, not only for enhancing communication, but also for documenting design rationale (if those communications can be captured effectively). In addition, MacMillan et al. stated that, "To the extent that [virtual] team members need to coordinate, better *mutual awareness* allows them to do so more efficiently, that is, with a lower communication overhead," (emphasis added) [97]. Everything that multi-user CAD and other communication tools can do to improve mutual awareness between team members will help them to coordinate and communicate more efficiently.

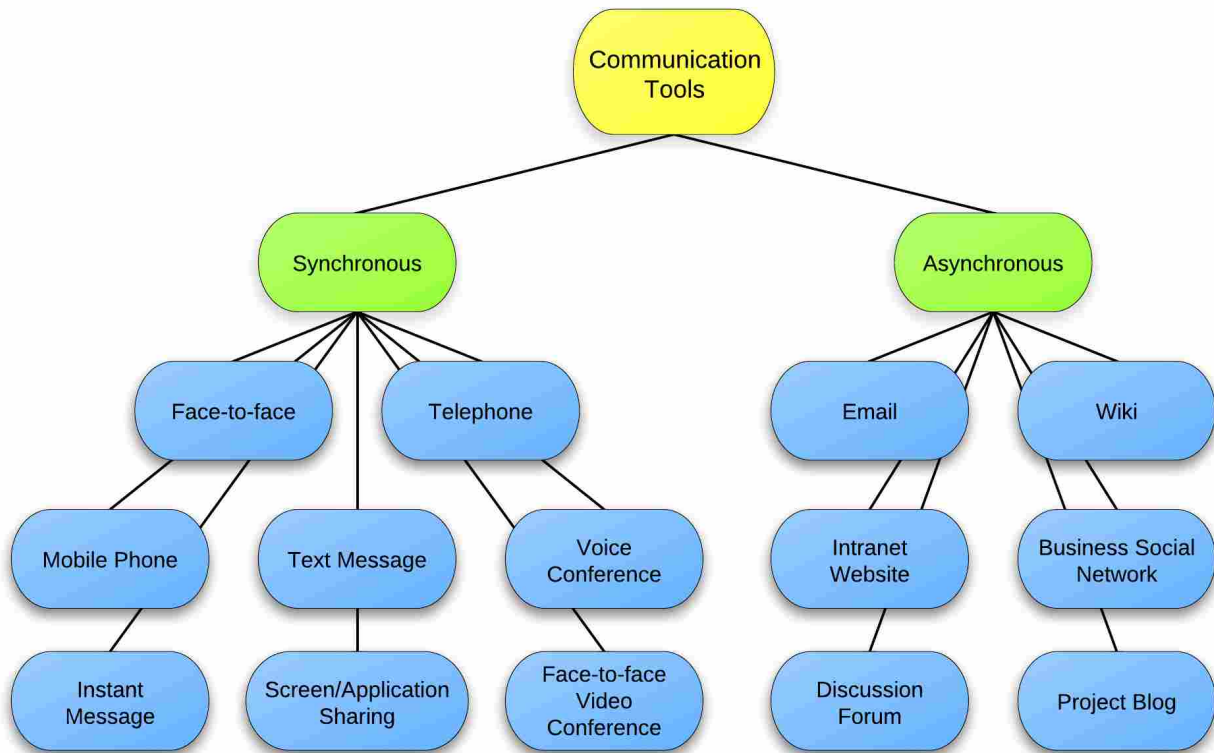


Figure 4.12: Communication tools.

Face-to-Face

Face-to-face communication is one of the most social forms of communication. Face-to-face communication allows for body language to be expressed and provides greater opportunities to cultivate social relationships. The body language of listeners can also give the speaker some information as to whether the listeners understand what is being said. Thus face-to-face communication can be a great facilitator of trust-building, as described above in section 4.2.6. Since face-to-face communication appears to be so valuable, we recommend colocating team members as much as possible to increase opportunities face-to-face communication and ad-hoc meetings.

Interestingly, video conferencing, which provides some of the visual communication that face-to-face communication provides, was rarely used, according to interviewees. Some participants reported that it was sometimes used for an initial team meeting at the start of a project to help build trust if a face-to-face meeting was not possible (e.g. due to travel expense). However, video

conferencing itself does not provide a lot of benefit above and beyond telephone since it does not help convey visual technical information as screen/application sharing does.

Email

Email appeared to be very valuable as an asynchronous communication tool for non-urgent communication, since email recipients can browse their emails at their convenience. Many interviewees also said they use when they want to have a written record of important communications and decisions.

We asked participants to estimate about how many work-related emails they receive per day. The results are shown in Figure 4.13. The median number of emails received per day was 50. Several participants mentioned receiving 10 to 25 auto-generated emails per day, so it is clear that many of the emails that engineers receive are directed at a wider audience than just that individual or even their team. 45% of study participants said that, in their opinion, they receive more emails than is appropriate.

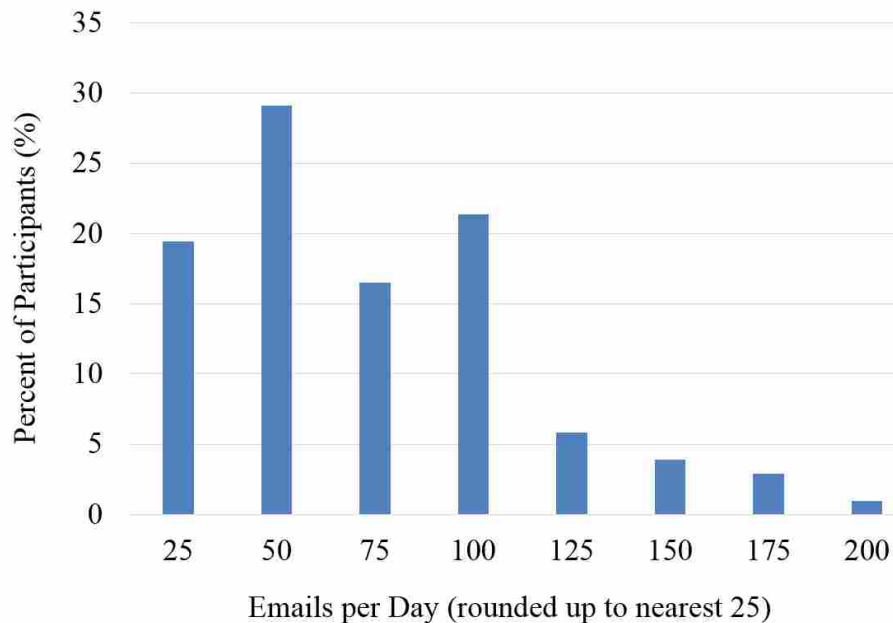


Figure 4.13: Number of emails received per day.

The number of emails received per day varied significantly by management level, with engineering team leads receiving about 15 more emails per day than technicians, and corporate-level engineers receiving about 25 more emails per day than engineering team leads (see Fig. 4.14).

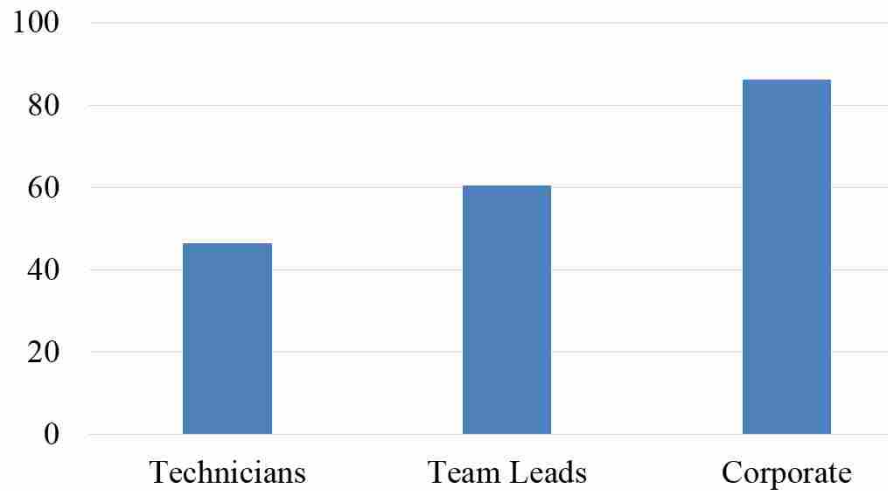


Figure 4.14: Average number of emails received per day by role.

An overload of emails can make it especially difficult to contact virtual teammates in a timely manner for urgent communications: One participant said, “When people are 8 time zones away you have to call or email them. And if they are swamped with emails that doesn’t help.”

Several interviewees mentioned that email is regularly misused. Sometimes people include too many people on an email (a similar problem as meetings). Sometimes, participants reported, the reason for this is that the email sender carbon-copies people for the sake of documenting their work and keeping their supervisors in the loop.

Another way that email is misused is when people don’t know when to start a new email thread or to use a different communication medium such as a phone call: One participant said, “Sometimes we use email when we should just get up and talk to someone. You can’t read tone in email... A lot of times you get into email wars whereas if you got together in the same room you could solve the whole thing in 10 minutes.”

In addition, sometimes people reply to all when a reply should only be sent to one person, or vice versa. Misuse of email is usually not a serious error, but a short training on email etiquette may help save some time and frustration and help people use this tool more effectively.

As for email's connection with collaborative engineering applications, perhaps appropriate emails that contain important design rationale could be directly linked to the product knowledge database or the design model.

Telephone

We asked participants to estimate about how much time they spend on the phone per day (including Voice-over-Internet-Protocol [VoIP], mobile phone, and conference calls). The results are shown in Figure 4.15: 40% of participants said that they spend about 10% of their time on the phone, and another 30% of participants said they spend 20% or more of their time on the phone. It is obvious that telephone is a dominant and critical communication medium in engineering industry.

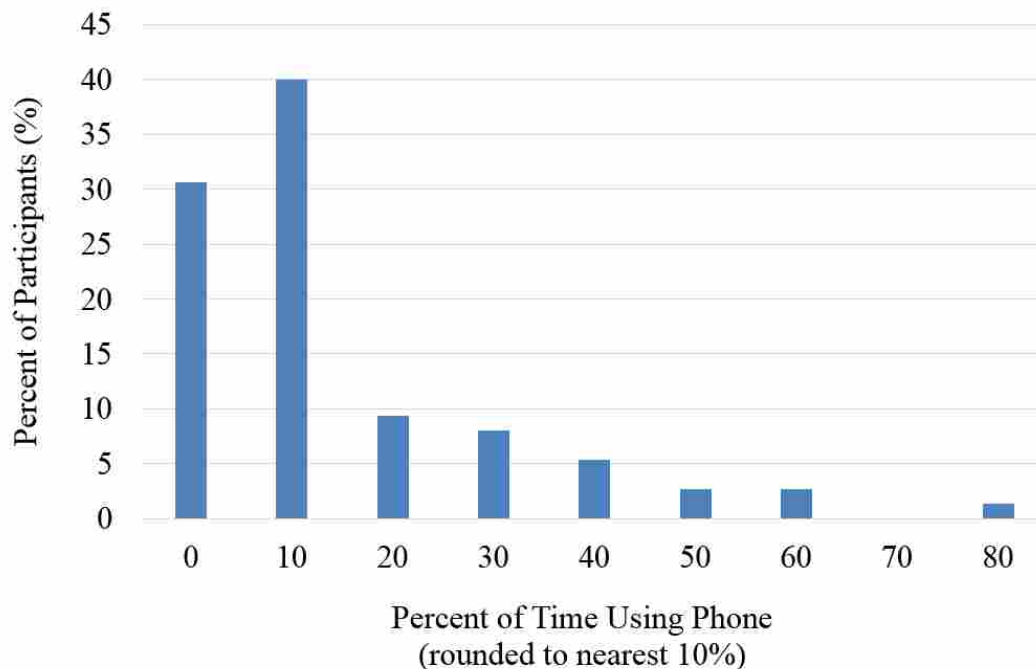


Figure 4.15: Percent of time spent on the phone.

In the case of language differences the phone was reported as sometimes being less effective than email (see section 4.2.6), but for same-language conversations, the phone was reported as being a very effective communication tool. People often used the phone for time-sensitive or urgent conversations, or to communicate something too complex, long, or sensitive to type in an email. Phone conversations are more difficult to document than written communication; thus it can be difficult to document design rationale communicated this way.

Screen/Application Sharing

Screen/application sharing appeared to be a core communication tool, especially for remote presentations, since it is very good for showing visual and technical information with a remote audience. It is also documentable if the specific tool being used (and the organization's policies) allows these meetings to be recorded.

Several participants reported technical issues with the actual use of this technology. One of the biggest issues is that it can be difficult to create or connect to a meeting, especially if the attendee is unfamiliar with the specific brand of tool being used. One participant said, "Some of it isn't completely intuitive, but once you've done it a few times, it's not too bad." Many participants reported spending an average of 10 to 15 minutes trying to get all of the attendees connected at the beginning of a meeting. In addition, firewall or other connection issues can hinder these meetings.

Issues were also reported with glitchiness and lag. Several people mentioned particular difficulties with lag when showing 3D models directly from the CAD system, so often times they had to resort to showing screenshots of the 3D model instead of the actual model. They also said that would be very valuable to show 3D models directly over a screenshare session if the experience could be smoothed out. Nysetvold et al. have proposed a method for implementing view-following directly into a multi-user CAD system using much less network bandwidth than traditional screen-sharing applications [14]. This could bypass screen-sharing entirely and provide a smooth communication experience to show 3D models remotely.

Instant Messaging

Instant messaging (IM) was used frequently at the companies we visited. Often IM was used to check if another person was at their desk before the sender of the message walked over for a face-to-face conversation. We suspect that IM will be a natural complement to multi-user applications, and that it may even be ideal if that type of a tool is integrated directly into multi-user CAD systems.

Wikis

Most of the engineering companies we visited had some kind of wiki for sharing information. Interviewees said that most people only used the company wiki intermittently and that many people were not even aware that a wiki existed. They said that most of the time when people have questions they just ask around the office, and they can usually find an answer fairly quickly or at least get contact information for someone outside their office who can answer their question. However, they also said that there were a small number of people who did use the wiki quite frequently, including remote workers who were not in a position to ask around the office.

Discussion Forums

Discussion forums were hardly used at all at the engineering companies we visited. We asked why this was so, since discussion forums are obviously such a prevalent medium for questions and answers on the internet. We were told that many questions people have in an engineering context are time-sensitive, so the person doesn't have time to wait for someone to notice the posted question and volunteer a response, especially since there are fewer potential responders at a company than on the internet. So in general, company-level discussion forums do not appear to have very much value at engineering companies.

Intranet Website

Responders reported that the company's intranet website was a valuable place to go to get general company information or find company-wide announcements, but it did not appear to be a tool that was used very actively or dynamically.

4.2.9 Technical Tools

To date, engineering applications have had limited collaborative capabilities. Product Life-cycle Management (PLM) systems can implement version control, user permissions, work flow, and central storage; and these systems have greatly improved engineers' ability to share technical files in a collaborative mode. Here are some of the difficulties that study participants reported with the technical tools they use:

- Difficulty translating data between engineering applications.
- Difficulty transferring and storing large files (for example, analysis result files) over a network.
- Difficulty finding data (i.e. "searchability").
- Difficulty managing versions, revisions, and configurations.
- Difficulty giving outsourcing teams access to necessary data stored in the PLM system.
- Usability issues (i.e. user interfaces, user experience); for example, hidden buttons and toolbars.
- Inefficient work flows.
- Application latency/speed.
- Poor or insufficient documentation and training materials.
- Lack of easy-access, simple-to-use help functions.
- Lack of instruction on best practices for using the technical tools.

- Difficulty trying to get measurements from a PLM tool.
- Only one user can work on each file at a time.
- Lack of markup capabilities.
- Some applications are not customizable enough (though many are).
- Application crashes (rare but very inconvenient).

On the subject of data translation, according to one study participant, automatic CAD translation tools can usually only translate about 80% of a CAD model correctly; the other 20% has to be done manually. That participant's full-time job was manually translating that "other 20%." Another participant reported data compatibility issues even between different minor versions of the same software application.

As the amount of virtual teaming increases, the need for more collaborative technical tools will also increase. Technical tools will need to evolve in multiple ways (collaborativeness, user experience, speed, functionality) to accommodate the increasingly intensive and virtually connected efforts of engineers.

The Need for Multi-User Engineering Applications

When asked what about their technical tools could be improved, several participants mentioned the need for more collaborative or multi-user capabilities. Here are some unsolicited quotes from interviewees to that effect:

- "The ability to share real-time information is invaluable. Communicating details of design, especially in assembly, is one thousand times faster and easier when both parties can look at the exact files when coordinating space and intent."
- "Teamcenter holds us back in certain ways in the way we share information between our sites. It's not truly instantaneous. One person can only work on one part at a time."
- "Nobody can do anything until [someone] checks his part back in. That was a huge problem. [Someone] would leave his part open all night, so it wouldn't upload and would cause huge problems."

- “It would be really really nice if you could use Excel in Google Docs mode,” (note that this was said in 2011, before Microsoft made a multi-user version of Excel).
- “I would like if there was a way to have multiple users collaborating on one [structural analysis] project.”
- “A lot of vendors have proprietary software that hooks in to [our system]. It would be nice if vendors could go in through the system in real time and update their package, but without giving them access to [our system].”

4.3 Conclusions

This chapter discussed the results of a set of surveys and interviews with several engineers and engineering leads from various engineering companies. It helped identify what collaboration challenges engineering industry currently experiences, how it copes with those challenges, how multi-user CAD will help address some of those challenges, and identify strategies for using multi-user CAD effectively in industry. 9 companies and 144 engineers participated from a variety of engineering industries. Most of these engineers were interviewed and surveyed.

We discovered that participants spend a median of 60% of their time collaborating (defined as working with others; not alone), and many people, especially team leads, spend as much as 70-90% of their time collaborating. If multi-user engineering applications such as multi-user CAD can help streamline some of that collaboration, that could result in a significant increase in the effective use of people’s work time.

We also discovered that team leads typically spend about 15% more of their time collaborating than other engineers. Thus the streamlining of collaboration will especially help team leads. In addition, if multi-user CAD can be used to increase the level of collaboration amongst team members, that could reduce the burden on team leaders in their role as a communication hub amongst their team.

On a scale from 1 to 10, 65% of study participants rated their team’s collaboration effectiveness at a 7 or an 8. This indicates that many people are happy with the collaboration on their team, but they still see room for improvement.

Communication was reported as being the biggest factor in making a team effective or ineffective. Examples of poor communication included not passing along important messages, incorrectly interpreting second-hand information, and using communication tools improperly. The researchers believe that multi-user CAD will help give team members access to up-to-date design data which will reduce the opportunity to miscommunicate 3D data using other means.

Participants reported that they spent a median of 30% of their time in meetings. 60% of respondents felt like they spent too much time in meetings. Formal meetings were reported as being moderately effective; whereas, ad-hoc meetings were reported as being very effective. Interviewees reported that having a prepared agenda, keeping the meeting on topic, and only inviting the necessary people to the meeting were keys to holding an effective meeting.

The median reported number of remote teammates that study participants worked with was 5. Interviewees revealed several challenges that are commonly encountered when working with a virtual collaborators, including language differences; time zone differences; security; tool incompatibility; and trust and social relationships.

Language differences can be especially difficult to deal with over the phone, and word-by-word translations do not always convey the communicator's true meaning. People in different time zones can be difficult to contact in a timely manner for urgent issues. And participants confirmed previous research which claimed that face-to-face project kick-off meetings can help improve trust and social relationships in a virtual team.

When different tools are used for the same purpose, data needs to be translated which often reduces the quality of the data. While multi-user CAD itself will not necessarily fix this problem, the additional collaborative opportunities it affords may increase the incentive for companies to try to use the same multi-user tools.

Security is a major consideration in engineering industry due to both competition and government regulations. Obtaining necessary export approvals can take more time than desired if the export official is unavailable. Since multi-user CAD naturally enables virtual teaming, it will be especially important for it to provide a way to appropriately protect sensitive data. This will be especially important for virtual teams with members in different companies and countries.

Based on the interviews it appeared that most of the companies that were visited used a matrix organizational structure. This type of structure seemed to be helpful in utilizing engineers'

talents both across the company and on a specific project; however, it also means that every engineer is accountable to two teams, which makes it difficult to manage priorities and schedules, and makes it more difficult to colocate team members for increased face-to-face communication and ad-hoc meetings.

In their discussion of communication tools, interviewees emphasized their use of face-to-face, email, telephone, and screen/application sharing; therefore, we believe that these communication tools are generally the most valuable and most frequently used amongst engineers at large engineering companies.

Face-to-face communication is valuable for building trust and social relationships. Since face-to-face communication is so valuable, we recommend colocating team members as much as possible to increase opportunities face-to-face communication and ad-hoc meetings. Video conferencing was rarely used except as a substitute for face-to-face trust building when a face-to-face meeting was not possible.

Participants receive a median of 50 emails per day which can make it difficult for them to respond to urgent emails in a timely manner. Several participants reported that email is often used improperly, leading us to believe that some kind of email etiquette training may be of benefit to teams that experience this problem.

Screen- and application-sharing tools were reported as being very valuable to engineers. Two main weaknesses of these tools were reported: It often takes 10 to 15 minutes to get everyone connected to a meeting, and these tools do not always show live 3D CAD data well (glitchiness and lag).

Several challenges or deficiencies were reported in regards to the technical tools (i.e. engineering applications) that participants used, including the following:

- Difficulty translating data between engineering applications.
- Difficulty transferring and storing large engineering files.
- Difficulty finding data.
- Difficulty managing versions, revisions, and configurations.
- Difficulty giving outsourcing teams access to necessary data.

- Usability issues (user interface, user experience).
- Poor or insufficient documentation and training materials.
- Lack of instruction on best practices for using the technical tools.
- Lack of markup capabilities.
- Only one user can work on each file at a time.

Several engineers specifically mentioned a desire for more collaborative, even multi-user, engineering applications. Technical tools will need to evolve in multiple ways (collaborativeness, user experience, speed, functionality) to accommodate increasing performance demands as well as the increasingly collaborative efforts of engineers.

4.3.1 Principles

Our investigation of technical collaboration with available tools, mostly single user and serial, would suggest that the following principles for improved synchronous collaboration have emerged:

- This study found that collaboration in technical product development involves 50% or more of available work time per day. Multi-user tools promote synchronous and thus immediate collaboration among a team, which for more complex products will improve the efficiency of team collaboration.
- Communication awareness is a key characteristic for product development effectiveness when decisions must be made by several technical specialists. Multi-user CAD architectures promote awareness through group decision processes.
- Because they address specific and immediate problems, ad-hoc meetings prove more effective than formal meetings. Multi-user CAD and other communication technologies provide additional opportunities for virtual ad-hoc meetings.
- Considering existing single-user product development collaborative tools, there is a need for multi-user collaborative engineering applications in industry.

CHAPTER 5. A STUDY OF MULTI-USER DESIGN TEAMWORK IN MINECRAFT

5.1 Method

To answer the question, “How will multi-user CAD impact design collaboration?”, I and my fellow researchers conducted a study of design collaboration in Minecraft to better understand design collaboration in multi-user CAD. We surveyed Minecraft users in an attempt to learn in advance how multi-user CAD will be used, what new challenges and benefits it may provide, how it will change collaborative design, and to offer insights into how virtual design teams might effectively build, communicate, and manage projects in a multi-user CAD design environment.

We also studied WesterosCraft, a particularly successful Minecraft organization devoted to creating a single model of the entire world of George R. R. Martin’s book *Game of Thrones*. Our investigation of WesterosCraft’s design paradigm was facilitated by their extensive, publicly-available documentation of their design process. We considered discussions and design documents recorded in the WesterosCraft forum and their published videos [31, 98].

We also compared the results of the Minecraft study with the results from the E-Handbook study to see how closely the results compare with engineering industry and look for similarities. The E-Handbook study investigated modern collaborative engineering practices at major companies in the United States through surveys and interviews, and was discussed in Chapter 4.

5.2 Results

This section discusses the results of the survey of Minecraft players, a qualitative discussion of some of the methods used by WesterosCraft, and comparisons between the results of the Minecraft and E-Handbook studies. It also discusses the postulated effects of multi-user CAD on team organization, planning, design concurrency, communication, and mentoring, based on the results of this study.

The online survey used a self-selected sample. We posted the survey link on a variety of Minecraft websites and forums. A significant portion of our sample came from WesterosCraft users, since the creator of WesterosCraft kindly agreed to promote the survey to them. Responses were collected over a period of several months. We obtained 165 complete or partial responses to our survey; respondents were free to answer whichever questions they chose, so the number of responses varied by question.

Our survey sample reflects a fairly diverse group of Minecraft users, many of whom have a significant amount of experience designing and building in Minecraft. Of the 138 individuals who gave their geographic region, 49 were from Europe, 73 from the U.S., and the balance from Canada, Australia, and Africa. Of 146 who gave their gender there were 17 females and 129 males. Comparing the responses to the previous two questions with the results of the 2008 (most recent published) National Science Foundation's *SESTAT: Characteristics of Scientists and Engineers in the United States* survey, we see general similarity with the demographics of U.S. based engineers [99]. 107 of 147 respondents said that English was their native language.

The typical respondent had been playing Minecraft for more than two years. About three-fourths of respondents played two or more times per week. On average, survey respondents reported spending more than half of their time in Minecraft building, as opposed to mining, fighting, or other. Respondents spent on average 31% of their playing time in multiplayer mode.

5.2.1 Team Organization

Team Size

Research indicates that the best team size is somewhere between three and seven members [84, 85], although this is typically given as the size of the smallest sub-team within a group. The E-Handbook study reported a reasonably close finding: The median reported team size (defined as “those with whom you work directly on a regular basis”) in engineering industry was nine people. We were interested to learn how many people can effectively work together in Minecraft. Minecraft players reported a median ideal team size of six players (see Fig. 5.1). This corresponds with Parker's [84] and Putnam's [85] findings.

Of 147 respondents, 47 said that they had participated in a team of 30 or more members. We also learned that large WesterosCraft projects have involved 40 or more individuals. Of those respondents who had participated in large builds, most reported very positive experiences: 33 respondents said that the best team they had ever played with had 30 or more members (see Fig. 5.1). The group of individuals who had experience working in a large team is referred to hereafter as the “large team subsample”.

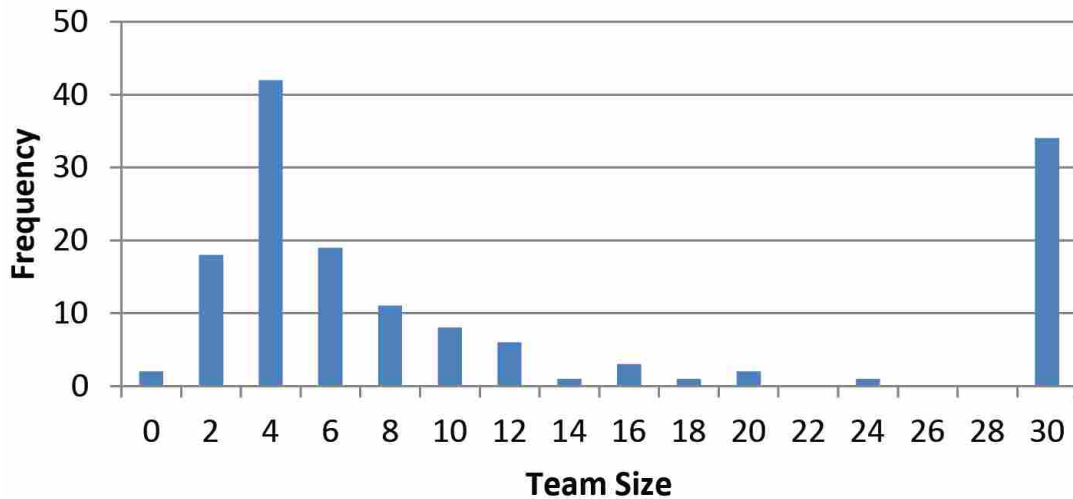


Figure 5.1: Size of the best team respondents have worked with in Minecraft.

Comments on working with a large team were almost always positive: “Most of the time, we are an absolute machine. With sometimes upwards of 40 builders at a single time, we can have the principle shape outlined of a massive castle in 30 minutes to 1 hour.” But one user commented on the complexity of such work: “Server builds [builds with many simultaneous users] are the worst. We plan them extensively to make sure there’s very little room for things to mess up with 20+ builders helping out, however 20+ people is still hard to manage even then.” However, the fact that almost all respondents reported positive experiences working with large teams offers hope that large-scale collaborative efforts can succeed in a multi-user environment. Groups of this size are likely to be organized into subteams, as in WesterosCraft.

Teammate Attributes

The survey also offers insight into the characteristics others look for or appreciate in their coworkers. 145 respondents answered the question, “What things are important for you when you look for a teammate?” The results are presented in Fig. 5.2.

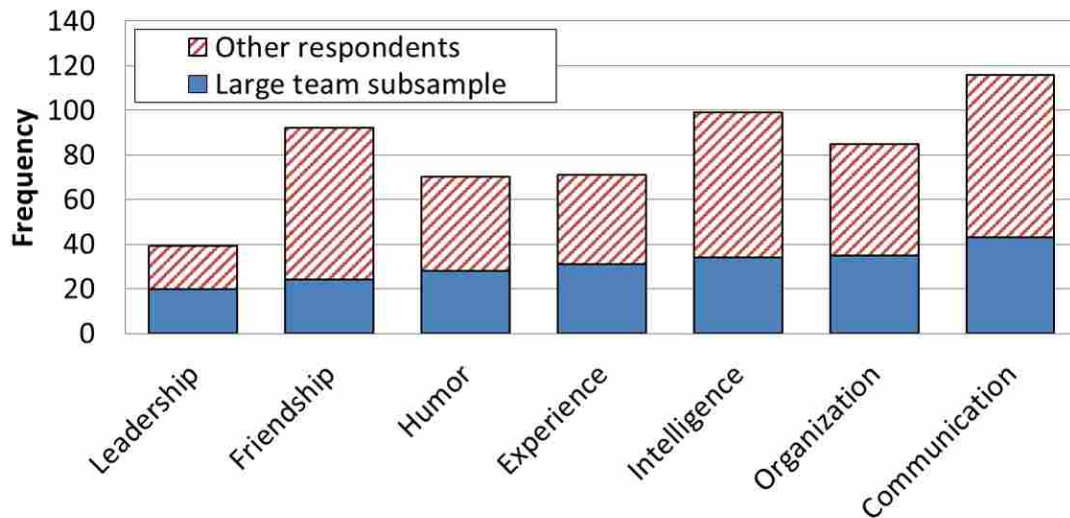


Figure 5.2: Desired teammate attributes.

From this result it is clear that communication is one of the most important attributes appreciated in teammates—even more so than intelligence or friendship. The importance of communication skills to the large team subsample suggests that improving communication could offer significant benefits. The E-Handbook study also found that communication was the most critical teaming attribute according to engineering industry professionals (see section 4.2.4). Various other authors, including Chan et al. and Dyer et al., have stressed that effective communication for new product development teams is crucial to achieving team goals [100, 101].

When asked to rate the importance of having worked with a team before and of knowing teammates in the “real world” on a scale from 0 (not important) to 100 (essential), having worked with a team before received an average rating of 62/100, while knowing teammates in the real world scored only 32/100. This indicates that experience working as a team in the past is considered very important for future teamwork. This may be tied partly to trust; one respondent commented, “Make sure you can trust your team members.” The large team subsample and the other respondents rated

working with a team in the past similarly, but the large team subsample rated the importance of knowing teammates in the real world only 15/100.

These findings run somewhat contrary to the literature regarding “virtual work” from the last ten to fifteen years and could indicate a significant shift in attitudes and/or abilities concerning virtual work. For example, Bailey and Kurland, Cooper and Kurland, Golden, and various others reported during the 2000’s that employees who “telecommuted” heavily reported lower job satisfaction, feelings of isolation from co-workers, and lower levels of trust [102–106]. The gaming experience, improved online communication tools, and new online social experiences might be a reason for this shift in attitudes, especially in the younger generation; therefore, young engineers might be more inclined to collaborate virtually.

5.2.2 Management

This section focuses on a few subtopics of management, but it is worth mentioning that WesterosCraft uses a variety of organizational techniques to facilitate its effective use of multi-user software, including extensive project planning, tiered leadership, multiple levels of peer review, tiers of editing permissions, and an application process for new members.

Team Management Structure

From survey comments it appeared that most, if not all, of the larger teams appeared to have a traditional pyramid style organization, having one main leader as well as sub-leaders overseeing sub-teams. Here are some of the comments:

- “We have a highly organized team structure, with a build leader, multiple team leaders and regular builders who do the ‘grunt work’.”
- “A hierarchy consisting of two admins at the top, several moderators and lots of builders.”
- “It was rather feudalistic, which is brutally effective. One head leader, three divided groups, each with one leader who reports to the main leader. Efficient and effective.”

- “We work best under a loose structure. The main project manager has the final say... Each moderator does thorough research and leads the build, but requires input from different specialists.”
- “Everyone technically did everything, but there was a fair amount of unofficial specialization.”

It appears that team management structures seem to remain fairly traditional within a multi-user environment. The E-Handbook study found that the majority of the engineering companies surveyed used a matrix organizational structure. This was commonly referred to as an integrated product team (IPT) structure. The IPT structure draws heavily from a pyramidal structure, but attempts to better integrate various engineering specialties. We suspect that this difference between Minecraft and engineering industry reflects the vastly greater variety of roles and specialization of expertise in engineering. Minecraft’s relative simplicity may make it easier to integrate specialties on an ad-hoc basis without the need to define roles as much as IPT structures tend to do.

Planning

WesterosCraft has a formal process for project proposals. To initiate a project, a user must prepare a proposal document that includes, among other things, diagrams of the proposed model, mockups of key areas and/or miniatures of the complete model, and background research demonstrating that the proposed design agrees with the relevant books and other material. These proposals are subject to peer review. Approvals by multiple moderators and the overall manager of WesterosCraft are required before construction can begin.

An emphasis on planning and top-down design is also evident in the large group subsample’s responses to the question “When building with others, what techniques do you use to divide work?” (see Fig. 5.3). Of the 49 individuals in this sample, 42 agreed that “We have a planning session before we start building” and 46 agreed that “The first thing we build is an outline of our desired structure,” while only 7 agreed that “We have little formal organization.” It may be tempting to believe that increasingly collaborative tools will allow for more unstructured design. However, the fact that planning, formal proposals, and top-down design practices are so extensively used in Minecraft suggests that they are just as essential for the effective use of multi-user CAD,

since CAD is even more complex and requirements and specifications are even more important for engineering design.

The following are several comments about the role of planning in effective Minecraft builds:

- “Before building we plan very carefully and everybody agrees before building”
- “One or two people plan, while the rest. . . do the manual labor like filling in floors.”
- “It requires work at the start, but once everything is decided by everyone, and we all agree on it, it’s fun from there.”
- “Everyone has specific tasks, whether exterior or interior, and details have already been specified by the research document.”
- “Some of the best creations I’ve participated in have been pretty much built on the spur of the moment. Existing plans can definitely help though if you have multiple people but I don’t let that limit the inherent creativity that comes at the spur of the moment.”

Task Management

Survey respondents mentioned a variety of task management strategies they used during actual modeling, including

1. continuous peer review,
2. dividing the model into sections with individuals assigned to those sections,
3. dividing the modeling effort into stages, and
4. management by wandering around.

Of these management strategies, we propose that (1) and (4) are better facilitated by multi-user applications than by single-user applications. Thus these management strategies may be more feasible in multi-user CAD than in contemporary, single-user CAD. Management mechanism (2) has been researched in the context of multi-user CAD by Red et al. using spatial constraint surfaces

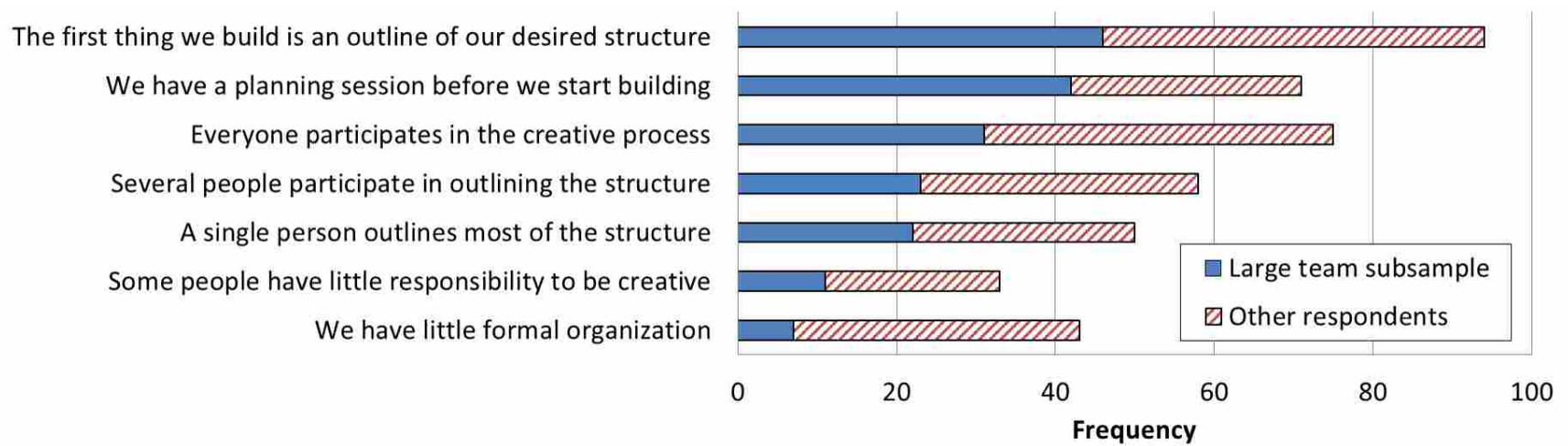


Figure 5.3: Techniques used by Minecraft teams to divide work.

[42]. The utility of dividing the model into sections in Minecraft supports the theory that this management technique could potentially be valuable in multi-user CAD.

Here are some comments from the Minecraft survey regarding task management:

- “You divide the build into sections and assign people to each.”
- “We work best under a loose structure. The main project manager has the final say, but does a lot of MBWA-style [management by wandering around] leadership.”
- “Iterative building loop, all parts of the building get reworked again and again for best results.”
- “Usually have everyone focus on one problem at a time, working as a single unit. Of course, specialization helps when it comes down to higher-level programming, higher-level design, or higher-level balance issues (for gameplay), but we try to have as many minds on each problem at a time.”
- “Peer review produces the best work.”

5.2.3 Collaboration

Although the primary benefits of team collaboration have become common knowledge, We present a few comments from Minecraft survey respondents to emphasize the benefits of teamwork in the multiplayer Minecraft environment:

- “Brainstorm fresh ideas that have never been thought of before, or expand on a pre-existing idea even more as a team and not as individuals.”
- “[Teamwork] provides more eyes to look at the creative process and therefore improve your build’s aesthetic appeal.”
- “For me it is essential to work in a team due to time constraints or the dauntingness of an overwhelming task.”

The E-Handbook study also highlighted the importance of collaboration: Non-leadership respondents reported spending an average of 55% of their time collaborating, while respondents

who were in a leadership role spent an average of 70% of their time collaborating. These findings highlight the need to facilitate collaboration and enhance the collaborative experience as much as possible. Based on our observations of Minecraft, We believe that modifying CAD (and potentially other applications) to support multi-user collaboration will help accomplish this.

Communication Tools

“It is somewhat difficult to transfer an image into other peoples’ minds, especially on a multinational server.” This comment, made by a Minecraft survey respondent, epitomizes the need for effective communication tools in a distributed teaming environment and agrees with research performed by Chan et al. [100].

The E-Handbook study identified several types of communication tools used in engineering industry (see section 4.2.8). Synchronous communication tools are those in which the communicators typically respond immediately to each other’s communication (e.g. telephone and face-to-face communication). Asynchronous tools are identified as those for which an immediate response is not typically expected. It may be useful to note that a multi-user application is itself a synchronous communication tool, since it conveys information about what others are doing in real-time, and it can complement other communication tools by allowing discussions to occur in a richer and more real-time context.

When asked about their collaboration tool usage and preferences, Minecraft survey respondents showed a very strong preference for voice chat (see Fig. 5.4). Despite the fact that text chat is built into Minecraft while voice chat requires a third-party program such as Skype [107] or TeamSpeak [108], even more users claimed to use voice chat (83%) than text chat (75%). One respondent noted, “Constant advice going through TeamSpeak.” Very few players used screen sharing (5%) or video chat (3%).

One respondent reported a weakness of text chat: “I don’t feel that using the in-game chat is very useful when building, as you have to stop what you are doing and type, which can take time if you are explaining something. You then don’t know if the player typing is typing, or has gone afk [away from keyboard].” The E-Handbook study discovered that 80% of the engineering professionals surveyed use text chat regularly at work, but many of them reported just using it to see if a coworker is at their desk before going to talk to that person face-to-face. We suspect that in

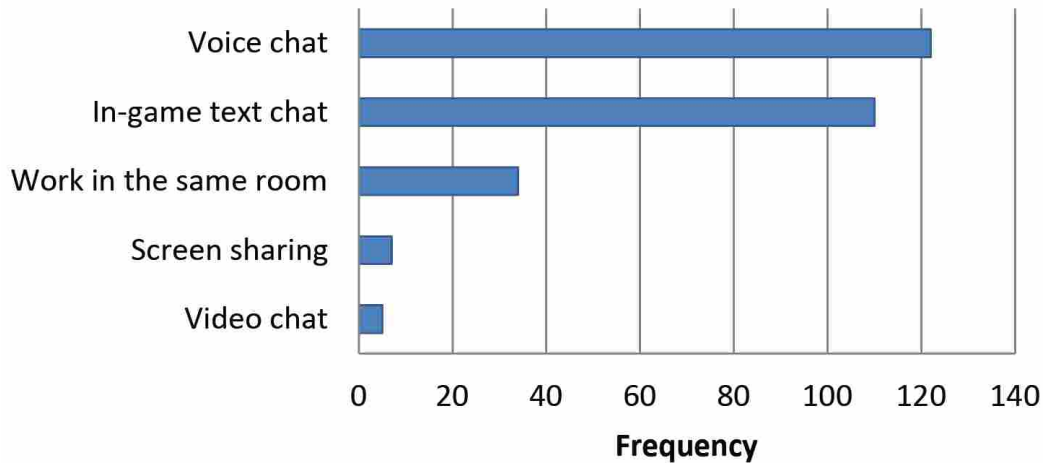


Figure 5.4: How Minecraft players typically communicate with their teammates.

multi-user applications text chat will mostly be used to communicate short pieces of information or questions, or it will be used to coordinate the use of a richer medium of communication such as voice chat.

In the Minecraft survey, when asked to rate the effectiveness of various communication methods on a scale from 0 (worst) to 100 (best), voice chat received a higher average rating (81/100) than working in the same room (71/100), text chat (50/100), screen-sharing (45/100), or video chat (42/100) (see Fig. 5.5). We suspect that the multi-user environment is largely responsible for users considering voice chat to be more effective than working in the same room, since the users have a shared context. If Minecraft players are all working in the same virtual context, it seems that vocal communication fills the greatest communication gap. If players are in the same room, but looking at their own screens anyways, then being in the same room apparently does not add much value to the collaborative experience. Some Minecraft players also commented that they preferred the privacy of working by themselves. The shared multi-user environment gives them the luxury of being able to do that.

We were also curious to ask which communication tools users wanted integrated into the Minecraft application (as opposed to using third-party communication tools). Almost all respondents (93%) said they would like voice chat to be integrated into Minecraft, and about one-third of respondents said they would like screen-sharing to be integrated (see Fig. 5.6). Although voice-chat is readily available by third-party applications, it seems users want voice chat to be integrated

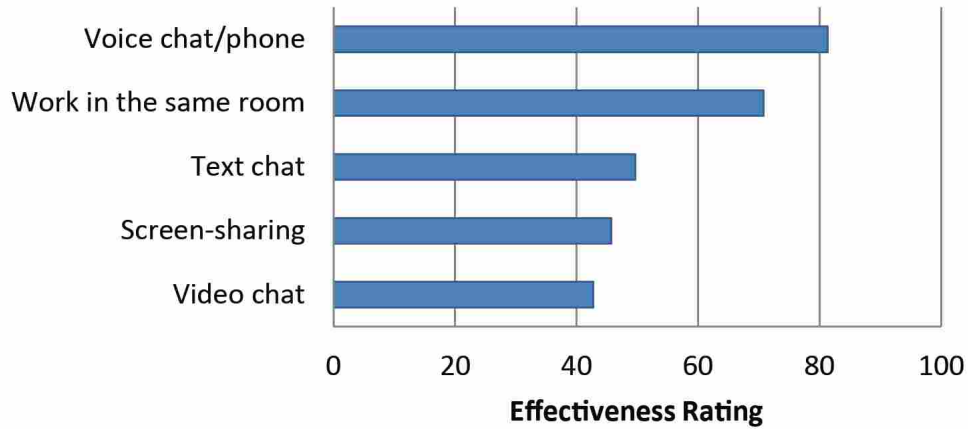


Figure 5.5: Average effectiveness ratings of communication tools (scale from 0 to 100).

for the sake of convenience. We propose that it would likewise be worthwhile to integrate both text chat and voice chat into multi-user CAD applications. Since a significant but minority proportion of surveyed Minecraft players would like screen-sharing to be integrated into Minecraft, We propose that it may also be worth integrating this feature into multi-user CAD applications. Nysetvold et al. have already proposed a method for integrating screen-sharing into multi-user CAD using much less network bandwidth than traditional screen-sharing applications [14].

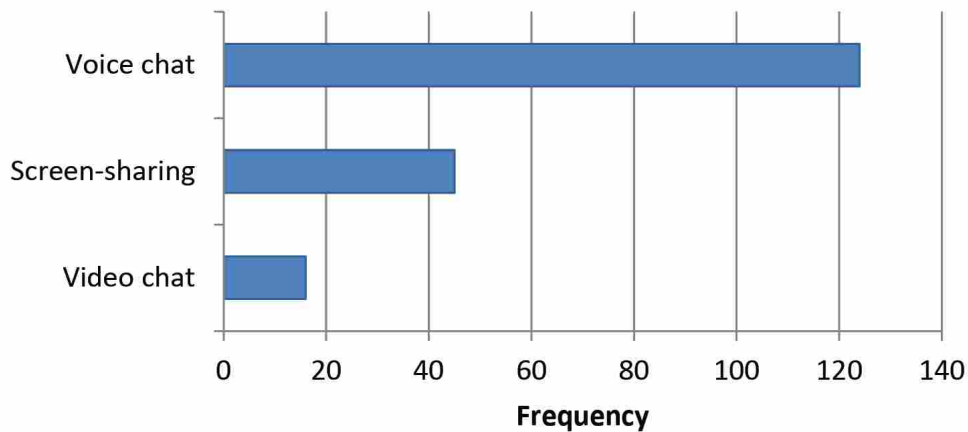


Figure 5.6: Which communication tools participants would like integrated into Minecraft.

In both the Minecraft and E-Handbook studies, video chat was not found to be a very popular communication tool. E-Handbook study respondents reported that, for engineering work, rich visual technical information is much more useful and necessary for collaboration than seeing

someone's face. They also reported that video chat or face-to-face meetings are most useful for establishing trust and building social relationships with each other; for example, a face-to-face or video chat meeting could be held with a distributed team during the kick-off phase of a project to build trust and familiarity amongst team members.

Minecraft users also tended to use online documentation and discussion tools such as wikis, forums, and discussion threads quite heavily. Some such websites contain hundreds of thousands of wiki articles, and have hundreds to thousands of active contributors [98, 109, 110]. However, the E-Handbook study discovered that these tools were used infrequently, overall, in engineering industry. Engineers claimed that wikis were more commonly used by virtual team members, and that local team members would more commonly ask around the office to find the information they wanted. We propose that online documentation such as wikis will be the most valuable to virtual teams, and therefore may be a natural complement to multi-user CAD.

Concurrency of Work

If multi-user CAD is adopted in industry, will engineers utilize the multi-user capability to work together in the same models at the same time (i.e. concurrently)? To help answer this, Minecraft survey respondents were asked about their practices and preferences in regards to concurrency of work (see Fig. 5.7). 72% of respondents said that their team either works together on the same model “usually all at the same time” or “mostly at the same time, but occasionally at different times.”

Respondents' preferences were even more in favor of concurrent modeling than their practices: 46% of respondents said that they prefer to work “usually all at the same time”, and another 39% said they prefer to work “mostly at the same time, but occasionally at different times.” This adds up to 85% of respondents who say that they prefer to work with their team on the same model at least “mostly at the same time”. This suggests that when multi-user CAD becomes commonly available many engineering professionals will likely utilize and appreciate the opportunity to work concurrently in a shared environment.

During their interviews a few of the E-Handbook study participants (engineering professionals) offered unsolicited comments that indicated their desire for more collaborative, perhaps even multi-user, software applications:

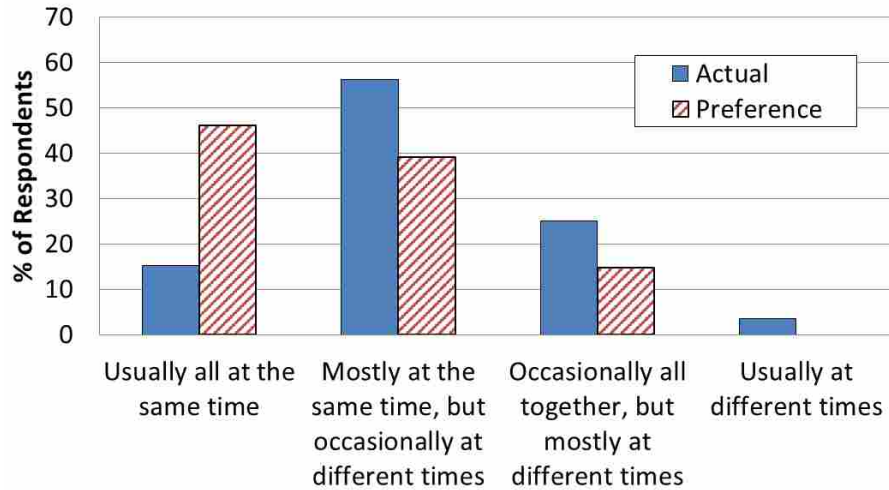


Figure 5.7: Concurrency of teamwork in Minecraft.

- “I would like it if there was a way to have multiple users collaborating on one project.”
- “It would be really, really nice if you could use Excel in Google Docs mode.”
- “One person can only work on one part at a time.”
- “Communicating details of design, especially in assembly, is 1000 times faster and easier, when both parties can look at the exact files when coordinating space and intent.”

From our observations, it seems that virtual teaming is much more predominant in the Minecraft community than it is in engineering industry. Thus, we propose that multi-user CAD will allow more distributed teaming to take place in engineering industry.

Mentoring

Minecraft participants were also asked, “What helps you become a better Minecraft builder?” The results are summarized in Figure 5.8. “Working with more experienced builders” and “practicing alone” were selected as the most popular methods for learning. Seeing videos or pictures of what other people have built was considered a slightly better learning method than watching videos of others building.

In their comments some Minecraft survey respondents mentioned the benefits of learning in a multi-user environment:

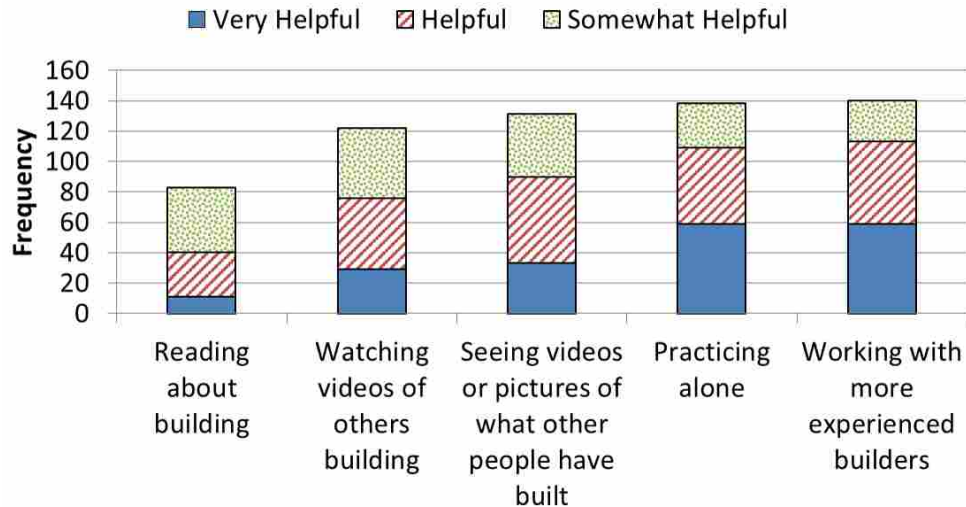


Figure 5.8: Effectiveness of learning methods in Minecraft.

- “...you learn 100x more about building and about Minecraft in general than when you’re alone.”
- “You can learn a lot if you have the appropriate teammate.”
- “Teamwork is a great way to improve as a builder through learning from others, and in turn sharing your information and tips with other, less experienced builders.”

Since multi-user CAD will allow users to see each others’ work in real-time, it will provide less-experienced engineers with more opportunities to observe more-experienced engineers work, thus supporting the learning method found to be the most popular among Minecraft players. Therefore we propose that multi-user CAD will enhance the learning and training of engineering design skills.

Teamwork Challenges

Minecraft survey respondents were also asked, “What becomes challenging when participating in a build with many other people?” (see Fig. 5.9). The first- and third-most selected responses were “While building, there are disagreements about details of the design,” and “At the start, everyone doesn’t agree on what to build,” respectively. One Minecraft respondent noted:

“Good teamwork in Minecraft is not an easy task. It is very important that the members of the team don’t easily get offended. Redoing each other’s work in some kind of iterative loop is a very good system, but requires good social skills.”

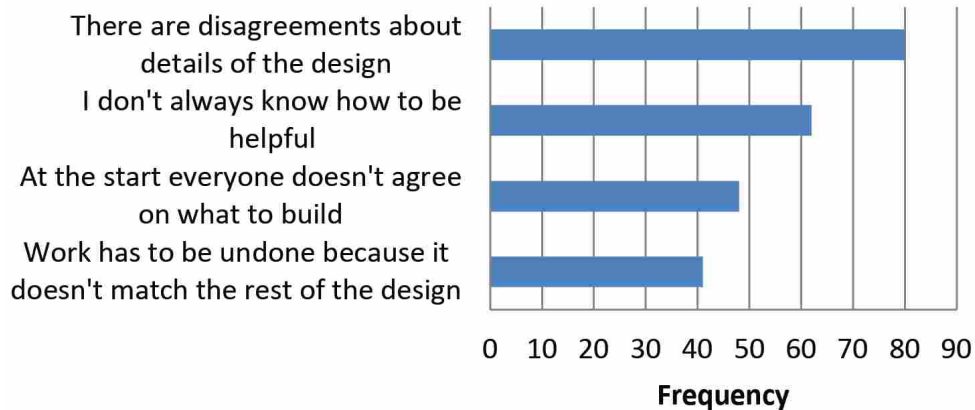


Figure 5.9: Teamwork challenges in Minecraft.

Disagreement and conflict can be good under the right circumstances [111, 112]. Teams without conflict may be suffering from other underlying problems. The fact that disagreements about design is the biggest challenge that players face indicates that their tools are supportive enough that they can focus on discussing the design, which is exactly the task they should be focused on.

The second-highest response was “I don’t always know how to be helpful.” This reflects the difficulty of managing a large team, and it is possible that this will become even more difficult in a multi-user environment. The fourth-highest response, “Too much work has to be undone because it doesn’t match the rest of the design,” was selected by 29% of users. We propose that although these challenges will indeed be faced in a multi-user environment, they would be even more difficult to deal with in a single-user environment.

5.2.4 Comparison of E-Handbook and Minecraft Studies

Table 5.1 provides a summary comparison of the findings of the E-Handbook and Minecraft studies.

Table 5.1: Comparison of findings from the E-Handbook and Minecraft studies.

	E-Handbook Study	Minecraft Study
Team Size	Median reported team size was 9 people.	Median ideal team size was 6 people.
Teammate Attributes	Communication skill was reported as the most important factor in team collaboration effectiveness.	Communication skill was the most desired teammate attribute.
Management Structure	Most companies used a matrix structure to optimally utilize a vast breadth of specialties across the organization.	Variety of structures, but the most common was a pyramidal structure.
Communication Tools	Face-to-face, phone, email, and web conferencing were the predominant communication tools.	VoIP (similar to phone) and instant message were the predominant communication tools.
Multi-User Capabilities	Several engineers reported a desire for multi-user capabilities in their applications.	Players strongly preferred working together at the same time.

5.3 Conclusions

A survey of 165 Minecraft players was administered and analyzed for the purpose of understanding the benefits and challenges of future multi-user CAD tools and the changes this would introduce to the design process. Overall the data strongly supported the need for and benefits of multi-user applications in a design context.

The findings demonstrated that it is possible for as many as forty designers to organize themselves and work together in a multi-user design environment. Although we have not directly compared multi-user and single-user Minecraft build times, it is apparent from our survey and observations that build times are greatly reduced when large teams of skilled Minecraft users collaborate simultaneously. Strong communication skills is the most desired attribute in teammates, both in a multi-user design environment and in engineering industry.

Survey results also indicated that Minecraft users found a high level of satisfaction from collaborating with teams of people they had often never met in real life. This runs counter to many published articles about virtual teaming in industry and may indicate that, despite previous indications to the contrary, engineers and designers working on virtual design teams might be able to expect a rewarding and fulfilling collaborative experience using multi-user tools.

A pyramid-type management hierarchy was observed to be an effective management structure in Minecraft teams with as many as forty players. The fact that planning, formal proposals, and top-down design practices are extensively used in Minecraft suggests that these practices will be essential for the effective use of multi-user CAD, since CAD models can be even more complex than Minecraft builds. Task management by continuous peer review and “management by wandering around” are facilitated by a multi-user environment. Dividing the model into sections was shown to be an effective way to manage project work in a multi-user environment.

Text and voice chat were used extensively in Minecraft, and almost all Minecraft users wanted voice chat to be integrated into the Minecraft application (text chat is already integrated). We propose that future multi-user CAD applications can enhance collaboration significantly by integrating these two tools into the CAD application. It may also be beneficial to integrate a screen-sharing tool into multi-user CAD applications. Minecraft players, most of which work on distributed (virtual) teams, preferred voice chat to working in the same room with their teammates. Although online documentation and discussion tools such as wikis and forums are not frequently used in engineering industry, these tools appear to be very valuable to dispersed Minecraft teams; therefore, these tools may be a natural complement to multi-user CAD when it is adopted.

From our observations, it seems that virtual teaming is very much more predominant in the Minecraft community than it is in the CAD engineering community. Thus, we propose that multi-user CAD will allow more distributed teaming to take place in engineering industry. Three-quarters of Minecraft survey respondents said that their team either works together on the same model at least “mostly at the same time”, and an even greater proportion of respondents said that working together concurrently at least most of the time is their preference. This suggests that once multi-user CAD becomes commonly available, engineering professionals both will want to utilize and will actually utilize multi-user CAD’s enhanced collaborative capabilities. Some engineering professionals who were interviewed during the E-Handbook study expressed, unsolicited, the need and desire for multi-user applications in engineering industry.

Survey results indicate that the most popular methods for learning in Minecraft are, first, working with more experienced builders, and second, practicing alone. The biggest challenges for Minecraft players appeared to be (1) disagreements about details of the design, and (2) that players didn’t always know how to be helpful. The fact that disagreements about design is the biggest

challenge that players face indicates that their tools are supportive enough that they can focus on discussing the design, which is exactly the task that they should be focused on. The second challenge, that players don't always know how to be helpful, reflects the difficulty of managing a large team, and it is possible that this could become even more difficult in a multi-user design environment.

Overall, the results suggest that multi-user CAD will likely be useful and practical in an engineering setting with several simultaneous contributors.

5.3.1 Principles

The following principles help us understand how a multi-user environment affects design collaboration:

1. Planning and communication are just as or more important in multi-user design as in single-user design.
2. In multi-user design, effective and accessible communication tools may be even more desirable than collocation.
3. In multi-user design, users strongly prefer working at the same time rather than working at different times.
4. Users gain a high degree of satisfaction and social validation working together virtually with people they have never met in person.
5. Multi-user design tools allow teams to focus on solving problems together.
6. Our observations of multi-user design in Minecraft were generally supportive of the proposed advantages of multi-user design.

CHAPTER 6. A METHOD FOR DETECTING LOCAL UNDO CONFLICTS IN MULTI-USER CAD

6.1 Method

This chapter provides a classification of the types of conflicts that can occur during the use of local undo in multi-user CAD, and describes a method for detecting those conflicts. This method can be used to prevent and warn the user about undo conflicts and to provide users with information about the cause of the conflict so they can collaborate to resolve it. This method has been successfully implemented and tested in BYU's NXConnect multi-user CAD prototype.

6.1.1 Classification of Undo Conflicts

A general classification of conflicts that can occur during local undo in multi-user CAD has been developed. This classification describes three primary types of conflicts: Feature-Self (Type A), Feature-Child (Type B), and Feature-Parent (Type C). In each of these types, the first word (i.e. Feature) refers to the feature being undone by the local user, and the second word (i.e. Self, Child, Parent) refers to a feature that was recently acted upon by another user, and describes its relationship to the feature being undone. Each of these three types of conflicts encompasses four subtypes as will be described later.

To better understand these types of conflicts, consider Figure 6.1. In this figure, the pre-operation state represents the state of a CAD feature at a certain point in time. When the local user performs a standard command, the feature then enters a post-operation state. A subsequent undo command by the local user will then attempt to revert the feature to its pre-operation state.

Table 6.1 shows three general types of operations that can be performed on a feature (create, delete, and edit). For commands in which the post-operation state exists (create and edit), undo can cause Feature-Self and Feature-Child conflicts since other users may have performed recent commands on that same feature or a child feature that rely on the post-operation state. For

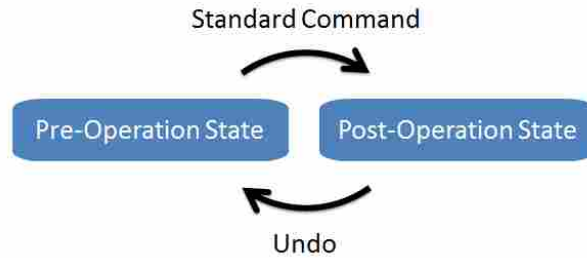


Figure 6.1: Standard and undo commands.

commands in which the pre-operation state exists (delete and edit), undo can cause Feature-Parent conflicts when the pre-operation state of the feature being undone relies on a parent feature that was recently modified by another user.

Table 6.1: Pre-operation and post-operation feature states.

	Pre-Operation State	Post-Operation State
Create	N/A	Exists
Delete	Exists	N/A
Edit	Exists	Exists

Feature-Self (Type A) Undo Conflicts

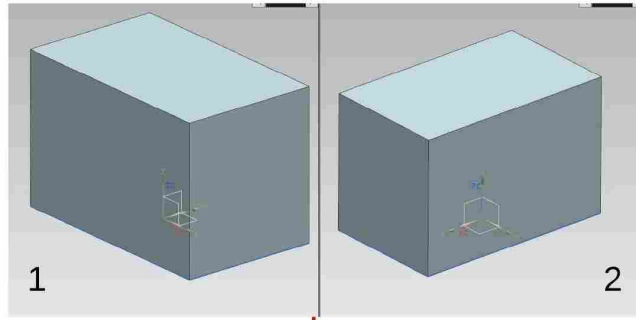
Feature-Self conflicts occur when the undo command operates on a feature that another user has modified since the command being undone was originally performed. Figure 6.2 demonstrates an example of this conflict type. Because User 2 edits the extrude that User 1 created before User 1 attempts his undo operation, a Feature-Self conflict occurs.

There are four subtypes of Feature-Self conflicts. Figure 6.3 describes these subtypes. Note that the example described in Figure 6.2 is a Type A.2 conflict.

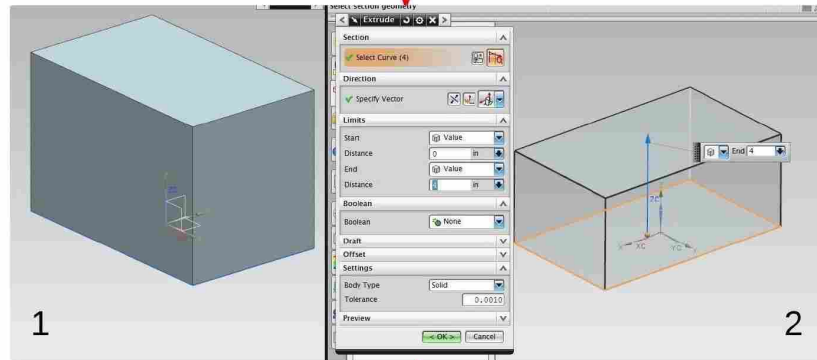
Feature-Child (Type B) Undo Conflicts

Feature-Child conflicts occur when other users have performed more recent commands that create or edit features that are dependent on the feature being undone. In Feature-Child conflicts,

User 1 creates an extrude



User 2 edits the extrude



User 1 tries to undo the creation of the extrude

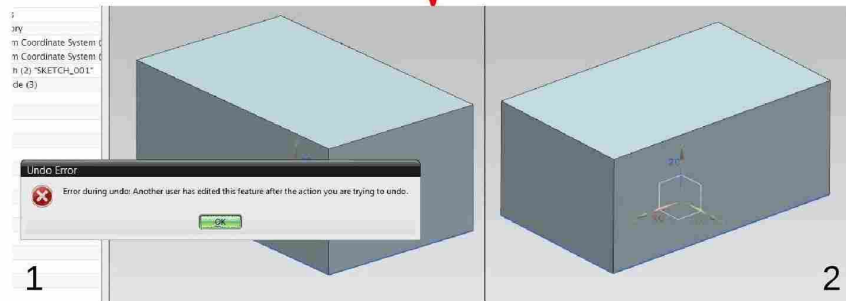


Figure 6.2: Example of a Feature-Self (Type A) undo conflict.

...but another user has recently _____ that **same** feature.

The intent of the undo is to _____ a feature...

	deleted	edited
delete	Type A.1	Type A.2
edit	Type A.3	Type A.4

Figure 6.3: The four subtypes of Feature-Self (Type A) undo conflicts.

it may be possible for the undo to occur, but the undo may negatively affect the recent actions or intent of other users. Figure 6.4 shows how this can occur. Since User 2 draws a sketch on the extrude User 1 creates, the sketch is dependent on the extrude. This results in a Feature-Child conflict when User 1 tries to undo the creation of his extrude.

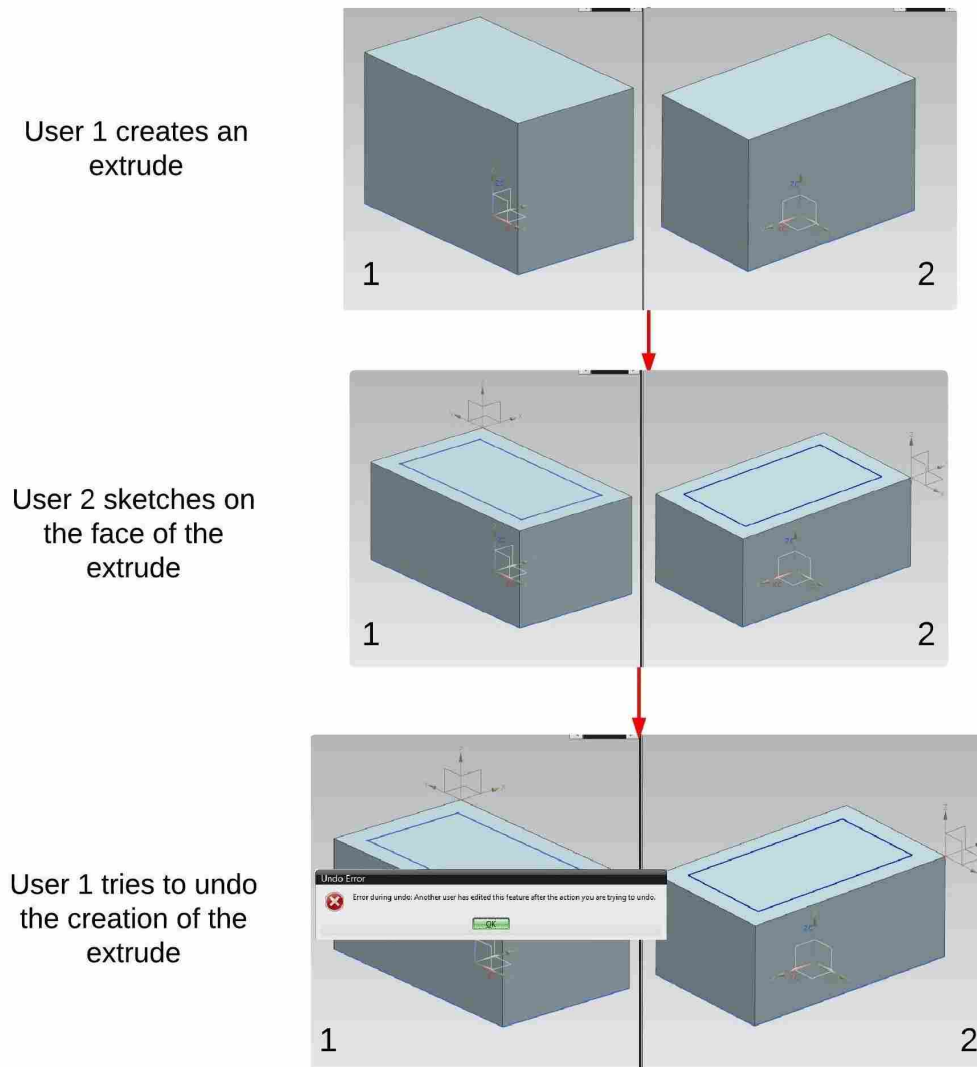


Figure 6.4: Example of a Feature-Child (Type B) undo conflict.

There are four subtypes of Feature-Child conflicts. Figure 6.5 describes these subtypes. Note that the example described in Figure 6.4 is a Type B.1 conflict.

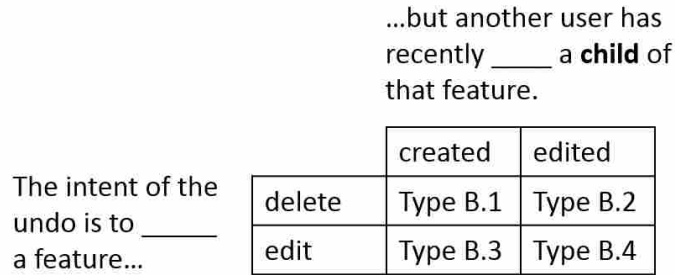


Figure 6.5: The four subtypes of Feature-Child (Type B) undo conflicts.

Feature-Parent (Type C) Undo Conflicts

Feature-Parent conflicts occur when other users have performed commands that affect the validity or intent of the undo being performed, because the necessary parent features either no longer exist or have been modified in a way that may conflict with the intent of the undo. Figure 6.6 shows how this can occur. When User 1 tries to undo his delete command, the original sketch his extrude was based on does not exist anymore; therefore his extrude cannot be un-deleted. This results in a Feature-Parent conflict.

There are four subtypes of Feature-Parent conflicts. Figure 6.7 describes these subtypes. Note that the example described in Figure 6.6 is a Type C.1 conflict.

Undo Conflict Subtypes and Suggested System Responses

Tables 6.2 and 6.3, below, describe the implications and suggested system response for each subtype of undo conflict. Table 6.2 is for Type A conflicts and Table 6.3 is for Types B and C. Note that for Types B and C, subtypes 1 and 2 can be treated similarly, and subtypes 3 and 4 can be treated similarly.

6.1.2 Method for Detecting Undo Conflicts

At a high level, the method for detecting undo conflicts is comprised of the following steps:

1. When users perform standard commands, set and send certain data.
2. When the user clicks undo, check for conflicts.

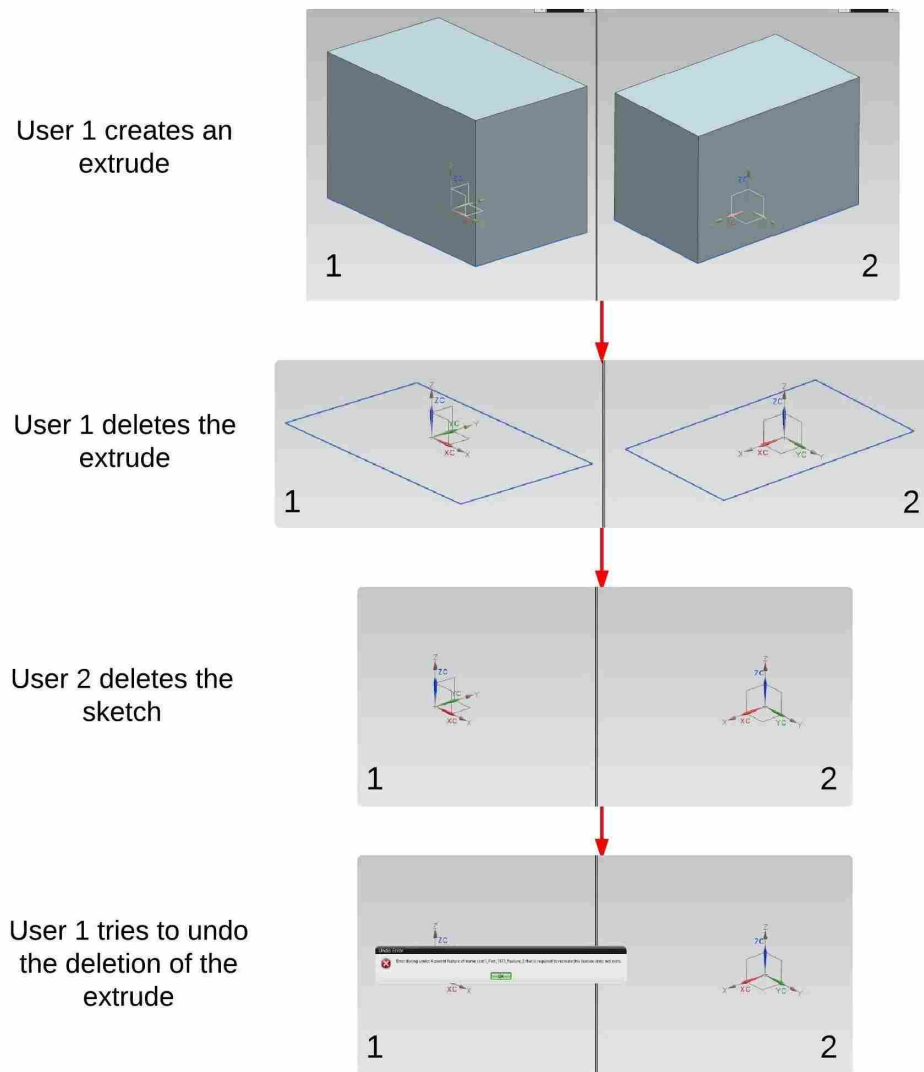


Figure 6.6: Example of a Feature-Parent (Type C) undo conflict.

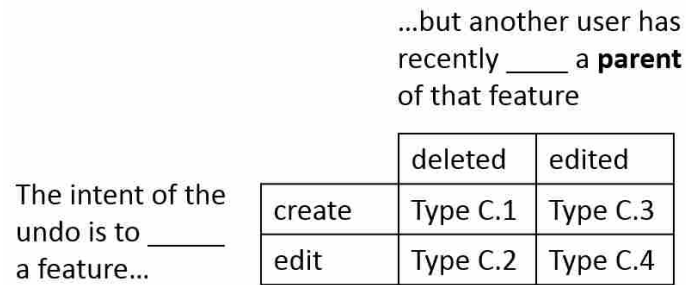


Figure 6.7: The four subtypes of Feature-Parent (Type C) undo conflicts.

Table 6.2: Implications and suggested system responses for Feature-Self (Type A) undo conflicts.

Type	Description	Implications	Suggested System Response
A.1	The intent of the undo operation is to delete a feature, but that feature has been deleted by another user.	The undo could be allowed since the result is the same as the other users intent.	“The feature you are trying to undo has already been deleted.”
A.2	The intent of an undo operation is to delete a feature, but that feature has been recently edited by another user.	Allowing the undo is syntactically possible; however, the deletion may interfere with the other users intentions.	“The feature you are trying to undo has been edited by User X. Your undo operation may conflict with User Xs design intent. Do you wish to proceed?”
A.3	The intent of an undo operation is to edit a feature, but that feature has been deleted by another user.	The feature could be re-created back to its pre-edit state; however, this may interfere with the other users intentions for deleting the feature.	“The undo operation is not allowed because the feature you are trying to undo has been deleted.”
A.4	The intent of an undo operation is to edit a feature, but that feature has been recently edited by another user.	The undo-edit could be allowed syntactically, but it may interfere with the other users intentions.	“The feature you are trying to undo has been edited by User X. Your undo operation may conflict with User Xs design intent. Do you wish to proceed?”

3. If any conflicts are detected:

(a) Block the undo.

(b) Tell the user why their undo was blocked and which user their command conflicted with.

4. After successful undo, revert certain data.

The local user’s command history is stored as shown in Berlage et al.’s diagram (see Fig. 6.8). In addition, to accommodate user commands that perform more than one operation (for example, deleting multiple feature at once), each Command object stores an ordered list of operations. Each operation corresponds with one modification performed on one feature (see Fig. 6.9).

Table 6.3: Implications and suggested system responses for Feature-Child (Type B) and Feature-Parent (Type C) undo conflicts.

Type	Description	Implications	Suggested System Response
B.1/B.2	The intent of an undo operation is to delete a feature, but child features have been (B.1) created or (B.2) recently edited since the most recent command performed by the local user.	Allowing undo to delete the feature will invalidate child features.	“The undo operation is not allowed because the feature you are trying to undo has child features that have been recently created or edited by User X.”
B.3/B.4	The intent of an undo operation is to edit a feature, but child features have been (B.3) created or (B.4) recently edited by another user.	Allowing undo to edit the feature may cause syntactic or semantic conflicts.	“The feature you are trying to undo has child features that have been recently created or edited by User X. Your undo operation may conflict with User Xs design intent. Do you wish to proceed?”
C.1/C.2	The intent of an undo operation is to (C.1) create or (C.2) edit a feature, but one of the features parent features have been deleted by another user.	Without the prerequisite parent features, the undo operation cannot put the feature back to its intended state.	“The undo operation is not allowed because the feature you are trying to undo depends on parent features that have been deleted.”
C.3/C.4	The intent of an undo operation is to (C.3) create or (C.4) edit a feature, but one of the features parent features has been recently edited by another user.	The feature may not be able to be restored to its intended state.	“The feature you are trying to undo has parent features that have been edited by User X. Your undo may or may not be successful in returning to its previous state. Do you wish to proceed?”

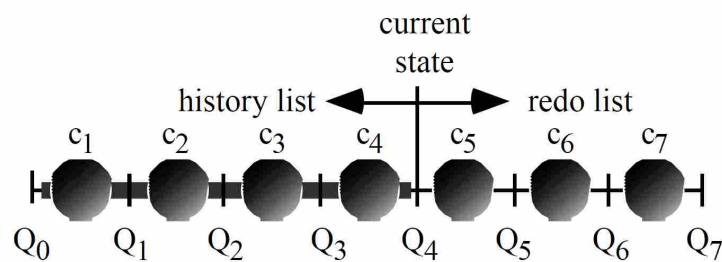


Figure 6.8: Operation history list (from Berlage [44]).

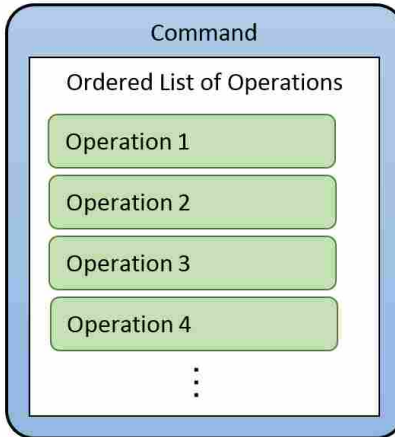


Figure 6.9: Each command stores an ordered list of operations. Each operation represents a change made to a specific individual feature.

Some metadata such as the last time that a feature has been edited and by whom each feature was last edited can also be stored to help give the user more information about why, when, and with whom each conflict occurred. The following sections describe how the various types of conflicts are detected. Refer to Appendix A for technical details about which data needs to be stored, when and how it is updated, and how that information is used to support this conflict detection method.

Detecting Feature-Self (Type A) Conflicts

To detect Feature-Self conflicts, an Operation Number (ON) is stored on the client. When a feature is created or edited, that feature is then tagged with the current ON. The ON increments after every create, edit, or delete command applied to the model from either the local user or other users. For example, if User 1 creates two features inside a new part, the first feature would be assigned an ON of 1, and the second feature would be assigned an ON of 2. If another user were to then edit the first feature, that feature would be assigned a new ON of 3. The ON is stored in the undo operation that gets made when the feature is created, so that when the user tries to undo, the undo operation's number can be compared to the feature's current ON to see if the local user was the last person to create or edit that feature.

Detecting Feature-Child (Type B) Conflicts

To detect Feature-Child conflicts, the ON of the child features is compared to the ON of the feature in question. If any of these child ONs are greater than the feature's ON, then someone else has edited the child feature since the local user's original action was performed. In this case the dependent children can be queried directly from the CAD feature, since the undo conflict check executes before the undo is actually performed.

Detecting Feature-Parent (Type C) Conflicts

To detect Feature-Parent conflicts, a list of the feature's parents needs to be stored in the operation data each time a standard command is performed in which the pre-operation state exists (i.e. edits and deletes; see Tab. 6.1), since the feature's previous parents cannot necessarily be queried directly from the CAD model after that feature has been edited or deleted. Before an undo occurs to go back to the prior pre-operation state, the model is checked to ensure that (1) the parent features still exist, and (2) the parent features have not been edited by other users after the command being undone was originally performed. The second condition can be checked by comparing the feature's ON with its parents ON.

Depth of Dependency Checking

Note that since grandchild/parent or great-grandchild/parent features might have been created, deleted, or recently edited by other users, a decision needs to be made as to how many levels of feature dependencies to check. In a large CAD model with many inter-feature dependencies, it might be impractical to check all levels of child features to see if any have been newly created or edited since the user's most recent command. The NXConnect implementation only checks the first level of dependent child features.

6.1.3 Successive Undo Commands by Multiple Users

This method can be extended to allow multiple users to collaborate to undo their conflicting commands in reverse order. Figure 6.10 shows two users collaborating to successively undo their commands to resolve User 1's undo conflict.

To support this, the pre-operation ON of a feature—that is, the ON the feature was assigned before the local user's command ever occurred—is also stored in the operation data. Then, after the local user performs a successful undo, the feature's ON can be reassigned to the value it had before the user performed his original command. This makes it look to other users as though the user's command was never performed, which allows them also to undo any commands they had previously performed on that feature.

In addition, successive undo requires that ONs be synced between clients. This can be accomplished using Hepworth et al.'s method by integrating the ON described in this chapter with Hepworth et al.'s operation sequence number [15].

6.1.4 Handling Multiple Operations in a Single User Command

It is possible for a single command to contain multiple operations: For example, a user selects three CAD features and deletes them in a single command, resulting in one command with three operations (one operation for each feature that was deleted). In this case, if any of the operations fail the undo conflict check, it may be most practical to disallow the undo of all operations in that command, since the user likely expects the entire command to succeed or fail.

6.1.5 Redo

Following an undo command, it is possible for other users to perform commands that will conflict with the local user's subsequent redo command. With some minor modifications, this method can detect redo conflicts as well as undo. Details of the changes required to support redo conflict detection are given in Appendix A, section A.2.6.

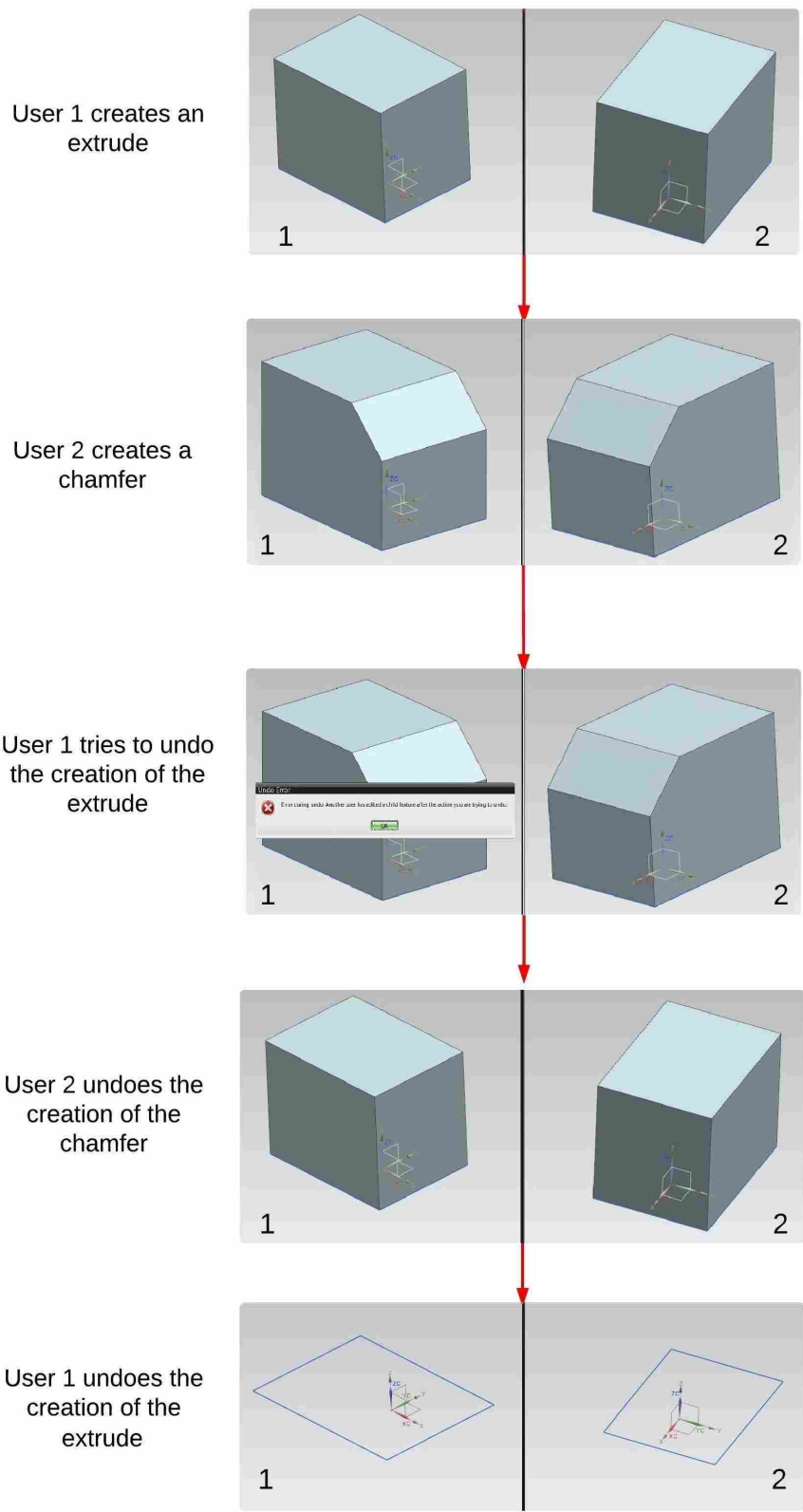


Figure 6.10: Example of successive undo commands.

6.2 Results

This undo conflict detection method was implemented in BYU's NXConnect prototype. The prototype succeeded at detecting the conflict types described above. The examples shown in Figures 6.2, 6.4, and 6.6 demonstrate the NXConnect prototype responding appropriately to each conflict type.

6.3 Conclusions

This chapter described two major accomplishments: (1) A description and categorization of local undo conflicts was established, and (2) a method was provided for detecting and warning about those conflicts. This method checks for both child and parent feature dependencies between features affected by the undo command and more recent commands from other users. It performs this dependency check prior to actually undoing the command, thus providing an opportunity to block or resolve the conflict before it actually occurs. This method has been successfully implemented and tested in BYU's NXConnect multi-user CAD prototype.

Detecting conflicts in local undo in multi-user CAD prevents model corruption and provides an opportunity for the users involved in the conflict to collaborate to resolve conflicts. This is an important technical challenge to address in multi-user CAD since undo is such a commonly used command, and since detecting conflicts in local multi-user undo is a prerequisite for resolving those conflicts. A method for detecting conflicts in multi-user CAD before an undo command is performed provides at least the following three benefits:

1. It will detect syntactic conflicts and some possible semantic conflicts before an undo or redo command is allowed. This will provide an opportunity for the system or the users to prevent or resolve the conflict.
2. It provides a way to inform the local user why their undo cannot or perhaps should not be executed and which other user performed the conflicting command. This allows the local user to collaborate with the user that performed the conflicting command to resolve the conflict. For example, the local user could ask the other user to undo their conflicting command so the local user's undo command can succeed, or both users could collaborate using standard (non-undo) commands to resolve the conflict.

3. It provides a way to inform the local user when another method or mechanism may be more suitable than local undo for accomplishing the user's intent. Examples of such alternative methods and mechanisms include standard (non-undo) commands; manual conflict resolution tools (commonly known as "diff" tools); and other types of undo mechanisms such as selective, regional, or global undo that could allow the local user to undo other users' conflicting commands before undoing their own command.

6.3.1 Principles

The following principles represent the accomplishment of the objective to develop a classification of types of undo conflicts in multi-user CAD and develop a method for detecting those conflicts:

- Undo conflicts in multi-user CAD can be classified into Feature-Self, Feature-Child, and Feature-Parent types, each with four subtypes.
- Feature dependencies can be queried to detect and block undo conflicts in multi-user CAD.
- When an undo conflict occurs, the user can be given information about why it happened and whose command conflicted with theirs.

CHAPTER 7. METHODS FOR ADMINISTERING DATA ACCESS IN MULTI-USER CAD

7.1 Method

This chapter describes methods that organizations could use to administer access to sensitive data in multi-user CAD. The following methods were developed:

- A method for administering spatial constraints in a multi-user CAD model.
- A method for restricting users from viewing a spatial region in a multi-user CAD model.
- A method for administering a discretionary access control (DAC) model, including restricting users from contributing to or viewing a multi-user CAD model.

These methods were developed and tested in BYU's NXConnect multi-user CAD prototype. A description of each of these methods follows.

7.1.1 Administering Spatial Constraints in a CAD Part

A method for administering spatial constraints access using spatial constraints has been developed. When a spatial constraint is defined and applied to a CAD part, users restricted by that spatial constraint are restricted such that they cannot create, edit, or delete features or geometry in the region restricted by the boundary. This will protect regions of the CAD model so that only those assigned or permitted to work in a given region are allowed to do so.

Please note the following terminology in the discussion that follows:

- *Manager*: A user who is allowed to administer (i.e. create, edit, delete) spatial constraints.
- *Spatial constraint*: An object in the multi-user CAD system that stores all of the information necessary to allow selected users to be restricted from editing or viewing CAD features located in a spatial region.

- *Boundary*: A set of constraint surfaces which together define the region of restriction.
- *Constraint surface*: A single 3D surface which can be used as a whole or partial definition of a boundary.

Principles

This method is comprised of the following principles and concepts:

- Multiple spatial constraints can be created, each with its own restricted spatial region and set of restricted users.
- Each boundary can be defined by one or more constraint surfaces.
- When a boundary is defined by more than one constraint surface, the restricted region can be defined by the intersection or the union of the allowed regions defined by each of the constraint surfaces.
- The user who defines a spatial constraint should have the option of deciding whether objects that intersect the boundary should be restricted.
- Administrators should be able to define whether datums are restricted, since in some cases it may only be desirable to restrict access to solid geometry and features.
- Constraint surfaces may be colored or otherwise identified to help restricted users and managers differentiate them from other surfaces.
- Each spatial constraint restricts one or more users or groups of users.
- Restricted users should be allowed to show or hide the constraint surfaces, since they may want to show it so they can see where they are allowed to work, or they may want to hide it so it does not obstruct their view.

The multi-user CAD client is responsible for enforcing these restrictions. It does so using Red et al.'s method for comparing geometry to a constraint surface [42]. If the geometry is selected that is inside the restricted region, the user will be warned that they do not have access to contribute

to that spatial region. Managers can administer spatial constraints via the multi-user CAD client; any changes they make get sent to other users via the multi-user CAD server.

Implementation in NXConnect

A working prototype that implements the principles defined in section 7.1.1, above, was built in NXConnect, BYU's multi-user CAD prototype. A screenshot of the NXConnect Spatial Constraints administration tool and constraints applied to a partial R/C car CAD model is shown in Fig. 7.1.

The prototype supports using datum planes as constraint surfaces to define the boundary; however, any type of surface could potentially be used. One or more datum planes can be selected to define the boundary of a spatial constraint. If more than one datum plane is selected, the Datum Interactions option lets the manager choose whether to allow users to work inside the region defined by the intersection of the regions allowed by the datum planes ("Restriction Priority") or the unity of the regions allowed by the datum planes ("Access Priority").

Each spatial constraint is assigned a unique color. That color is assigned to the datum planes that are selected to define the boundary. The direction of the datum plane's normal axis points in the direction that users are allowed to work on. A datum axis of the same color is created by the spatial constraints algorithm to show restricted users which side of the datum plane(s) they are allowed to work on. In Figure 7.1, the boundary of the spatial constraint called "Front Suspension" is represented by the blue datum plane and datum axis, and the boundary of the spatial constraint called "Rear Suspension" is represented by the green datum plane and datum axis.

Figure 7.2 shows the NXConnect Options form that gives users the ability to show or hide spatial constraints that apply to them and/or other users.

The definitions of the spatial constraints are stored in the NXConnect database. Figure 7.3 shows the structure of this database table.

7.1.2 Restricting Users from Viewing a Spatial Region in a CAD Part

A method has also been developed for restricting users from viewing a spatial region in a CAD part. To support this method a boolean option was added to the Spatial Constraints adminis-

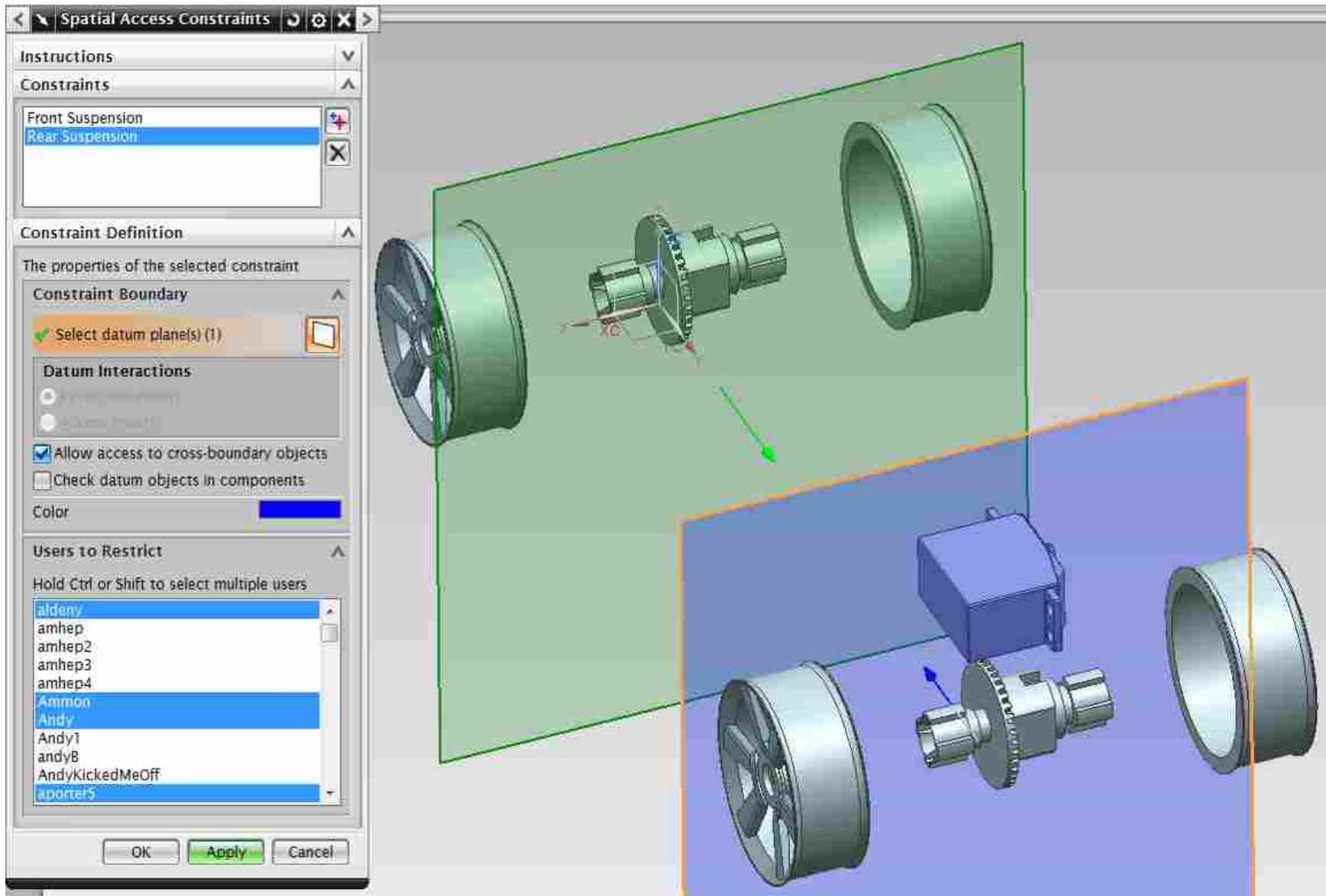


Figure 7.1: Spatial constraints in a partial R/C car CAD model.

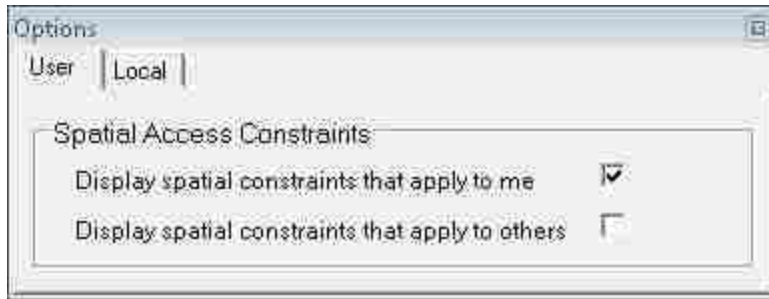


Figure 7.2: Constraint surface hide/show options.

	Column Name	Data Type
	PartID	Int
	Name	nvarchar(200)
	Users	nvarchar(MAX)
	ConstraintSurfaces	nvarchar(MAX)
	Color	tinyint
	AllowObjectsThatIntersectBoundary	bit
	IsRestrictionPriority	bit
	RestrictDatumObjects	bit
	HideRestrictedObjects	bit

Figure 7.3: NXConnect spatial constraint database table.

tration form called “Hide restricted objects,” as shown in Figure 7.4. This can be used to make a spatial constraint block viewing in addition to blocking editing. A boolean column was also added to the database table to store this information (refer to Fig. 7.3).

This method still stores the entire CAD part on the client, but whenever a new feature is received by other users the client uses the spatial constraint algorithm identified by Red et al. [42] to determine if the user is allowed to view that feature. If the user is not allowed to view that feature, then the feature is hidden. The user is also blocked from un hiding those features.

7.1.3 Administering Access Using a Discretionary Access Control Model

In multi-user CAD, a Discretionary Access Control (DAC) model gives managers and teams direct administrative control over who has access to the team’s CAD parts, and what level of access those people have.

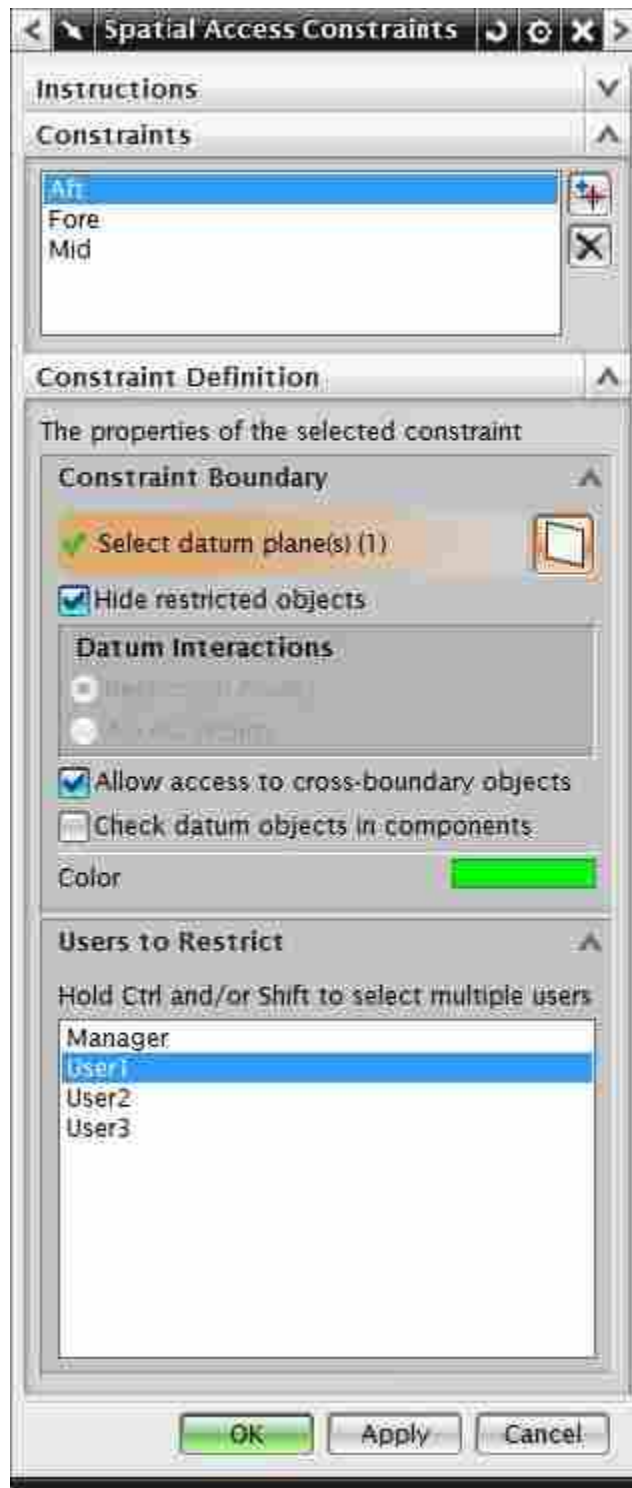


Figure 7.4: Spatial constraints administration form.

Principles

This method is comprised of the following principles and concepts:

- One or more access levels are defined (e.g. no access, view access, edit access, administration access).
- A default access level is defined for each CAD part that determines the access level for anyone in the organization who is not listed as having a specific access level in that part.
- One or more users can be assigned to specific access levels for that part.
- A manager has the ability to modify the access levels assigned to users.
- Each CAD part needs at least one manager assigned at all times.

When a user attempts to perform a certain action on a CAD part (e.g. view, edit, etc.), the system checks if that user is assigned a certain access level. If no specific assignment has been made for that user, the default access level is used. If the user attempts to perform an action that he does not have access to perform, the system rejects the action and warns the user. When this happens and the user is warned, it may also be helpful for the system to inform the user who they can contact to gain the access they need.

Implementation in NXConnect

A DAC model was implemented in BYU's NXConnect prototype. The DAC administration form is shown in Figure 7.5.

The access levels implemented include Manager, Contributor (i.e. editor), Viewer, and No Access. A user assigned to "No Access" cannot view, edit or administer access for a CAD part. A user with "Viewer" access can view a part, but cannot edit or administer access levels for that part. A user with "Contributor" access for a part can edit and view the part, but cannot administer access levels for that part. A user assigned as a "Manager" for a part can edit and view that part, can administer access levels for that part, and can administer spatial constraints.

Part Access Permissions

Part Access Level: Private Access

- Any User Can Manage
- Any User Can Contribute
- Any User Can View
- Private Access

Individual user access levels for this part

Username	Access Level
admin	Manager
Ammon	Manager
Andy	Contributor
apridis	Contributor
Brett	Contributor
dan	Viewer
dfrench	Viewer

Add User:

Delete Selected User | Add From List...

OK | Cancel

Figure 7.5: DAC part access administration form.

In Figure 7.5, the “Part Access Level” option defines the default access level of the part. The default access level is the access level that is assigned to all users in the organization who have not been assigned a specific access level for that part. The four options are:

- *Any User Can Manage*: All users in the organization are considered Managers in this part unless otherwise specified.
- *Any User Can Edit*: All users in the organization are considered Contributors in this part unless otherwise specified.

- *Any User Can View*: All users in the organization are considered Viewers in this part unless otherwise specified.
- *Private Access*: All users in the organization are considered Viewers in this part unless otherwise specified.

Specific users can be assigned higher or lower access levels for a part than the part’s default access level. In Fig. 7.5, several users are listed with their assigned access levels. These access levels can include the roles of Manager, Contributor, Viewer, or No Access, similar to the default access level options for parts.

To support the DAC system discussed above, four new data columns were added to the Parts table in the NXConnect database (see Fig. 7.6). The added data columns are as follows:

- *DefaultAccessLevel*: The default access level of the part.
- *Managers*: The usernames of Managers of this part.
- *Contributors*: The usernames of Contributors of this part.
- *Viewers*: The usernames of Viewers of this part.

	Column Name	Data Type
	PartID	int
	PartName	nvarchar(450)
	Description	nvarchar(200)
	Units	int
	DefaultAccessLevel	tinyint
	Managers	nvarchar(MAX)
	Editors	nvarchar(MAX)
	Viewers	nvarchar(MAX)

Figure 7.6: NXConnect Parts database table definition.

7.2 Results

The methods described above were implemented in NXConnect and tested using demonstrative CAD models to prove that they worked as intended.

7.2.1 Administering Spatial Constraints in a CAD Part

This method was successfully used to administer spatial restrictions in a multi-user CAD environment. The method and principles described above were tested using a simplified rocket CAD model. The spatial constraint administration tool successfully created constraints based on input parameters to control which users could access which spaces of the model.

Figure 7.7 shows a manager setting up spatial constraints in a simplified rocket CAD model. First the manager sets up the datum planes to be used by the spatial constraints. Second, he creates a constraint called ‘Aft’ to restrict User 1 to working on the rear section of the rocket, and chooses the datum plane on the left (that also points to the left) as the boundary definition. Third, he creates a constraint called ‘Mid’ to restrict User 2 to working on the middle section of the rocket, and chooses two datum planes that point inward as the boundary definition. Finally, the manager creates a constraint called ‘Fore’ to restrict User 3 to work on the area around the nose cone, and chooses the datum plane on the right (that points to the right) as the definition of that constraint’s boundary.

Figure 7.8 shows the views of the four users and the results of their initial actions. Now that the manager has assigned the spatial constraints, they have been removed from his view (as shown in the upper-left pane) because none of them apply to him. He attempts to modify the nose cone component and is allowed to do so.

The upper-right pane of Figure 7.8 shows User 1’s view. User 1 sees the green datum plane that represents the boundary of the ‘Aft’ constraint. He sees that the green datum axis points to the left, so he knows he is restricted to work on the left side of the green datum plane. He attempts to modify the nozzle of the rocket and is allowed to do so.

The lower-left pane of Figure 7.8 shows User 2’s view. User 2 sees the two blue datum planes that represent the boundary of the ‘Mid’ constraint. He sees that the blue datum axes point inwards, so he knows that he is restricted to work on the middle section of the rocket. However, he tries to modify the skin (highlighted orange in User 2’s view) and is blocked because when the manager created the ‘Mid’ constraint, he chose to restrict access to objects that intersected the constraint boundary, which the skin does.

The lower-right pane of Figure 7.8 shows User 3’s view. User 3 sees the green datum plane that represents the boundary of the ‘Fore’ constraint. He sees that the green datum axis points

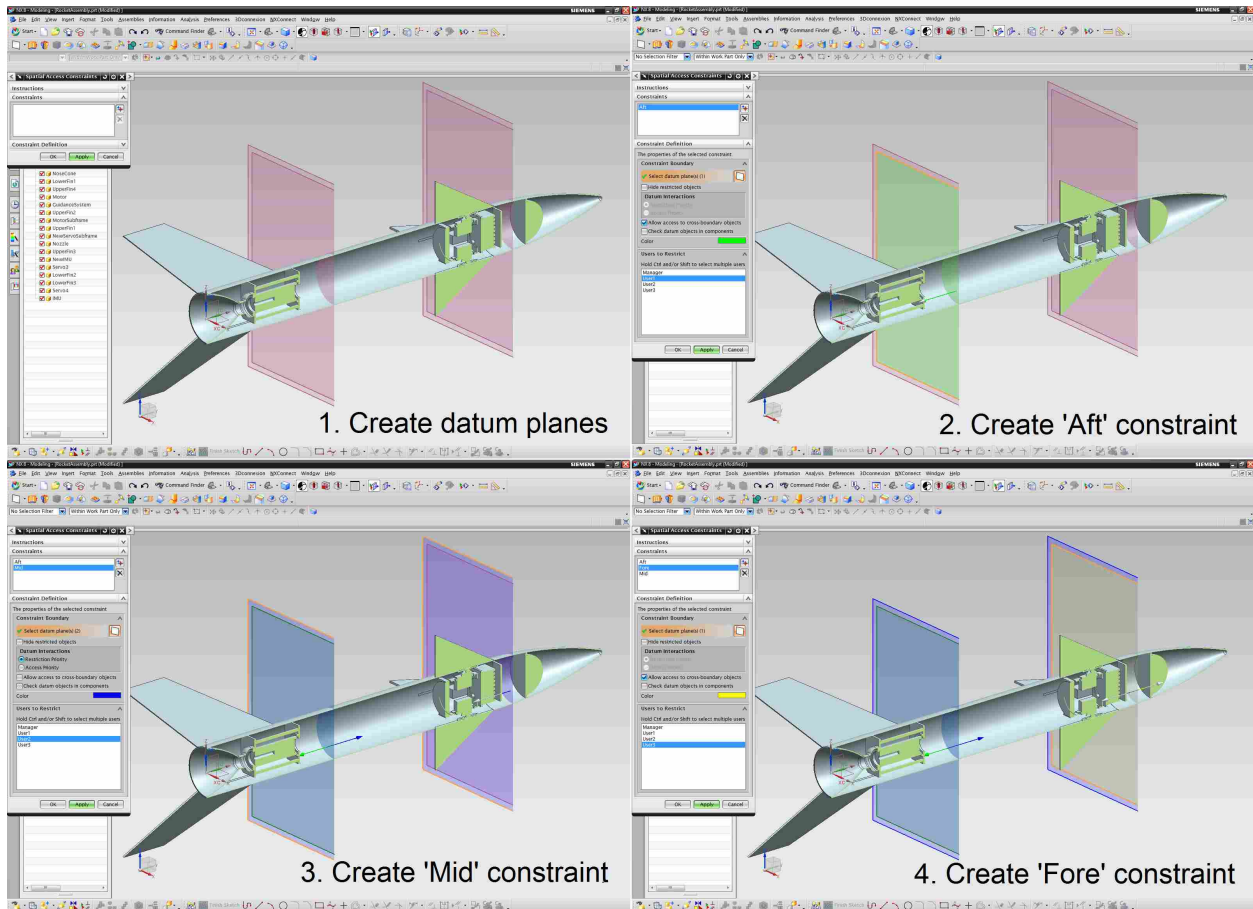


Figure 7.7: A manager creates three spatial constraints and assigns certain users to be restricted by each constraint.

points to the right, so he knows that he is restricted to work in the area around the nose cone. Despite this, User 3 tries to delete one of the lower fins at the aft of the rocket. His action is rejected by the system.

This scenario shows the successful administration and application of spatial constraints in a simple scenario. This method could be used in industry to effectively administer spatial constraints, thus limiting access to edit a CAD model to appropriate teams and individuals to protect the model from unauthorized changes.

7.2.2 Restricting Users from Viewing a Spatial Region in a CAD Part

Figure 7.9 demonstrates the same spatial constraint setup as shows in section 7.2.1, but this time the manager has enabled the “Hide restricted objects” option for each of the spatial

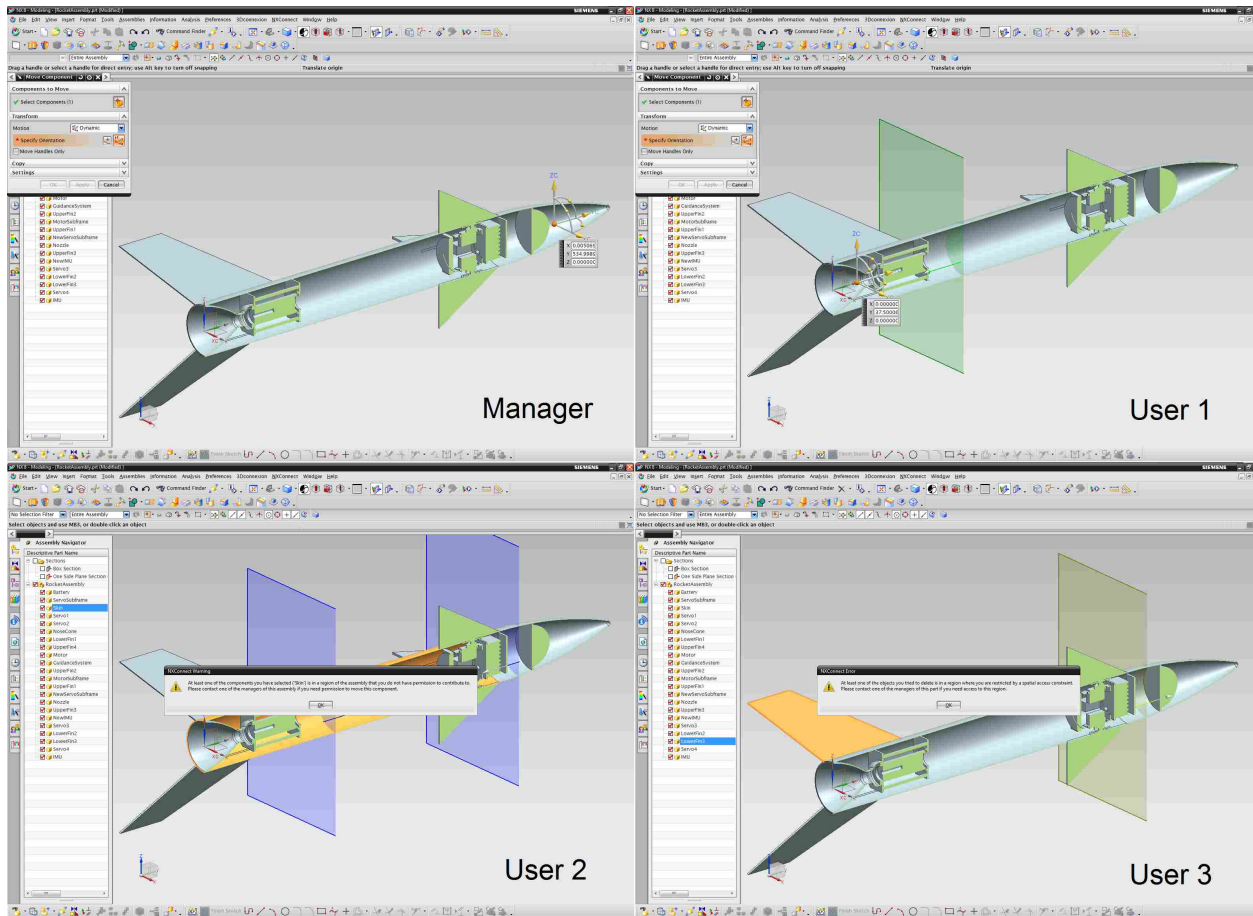


Figure 7.8: User views after being assigned to spatial constraints, and the results of their initial actions.

constraints. Each client stores the entire CAD model on their client, but all restricted objects are hidden, and the users can only see features they are allowed to view and edit. The users are still connected to the same multi-user session, so, in this scenario, the manager would see any changes made by the other users. If any of the other users' unrestricted regions overlapped with each other, they could collaborate and work together in those overlapping regions.

In addition to protecting the model from unauthorized changes, hiding restricted objects in spatial regions could be very valuable in industry for protecting unauthorized users from viewing sensitive designs and intellectual property.

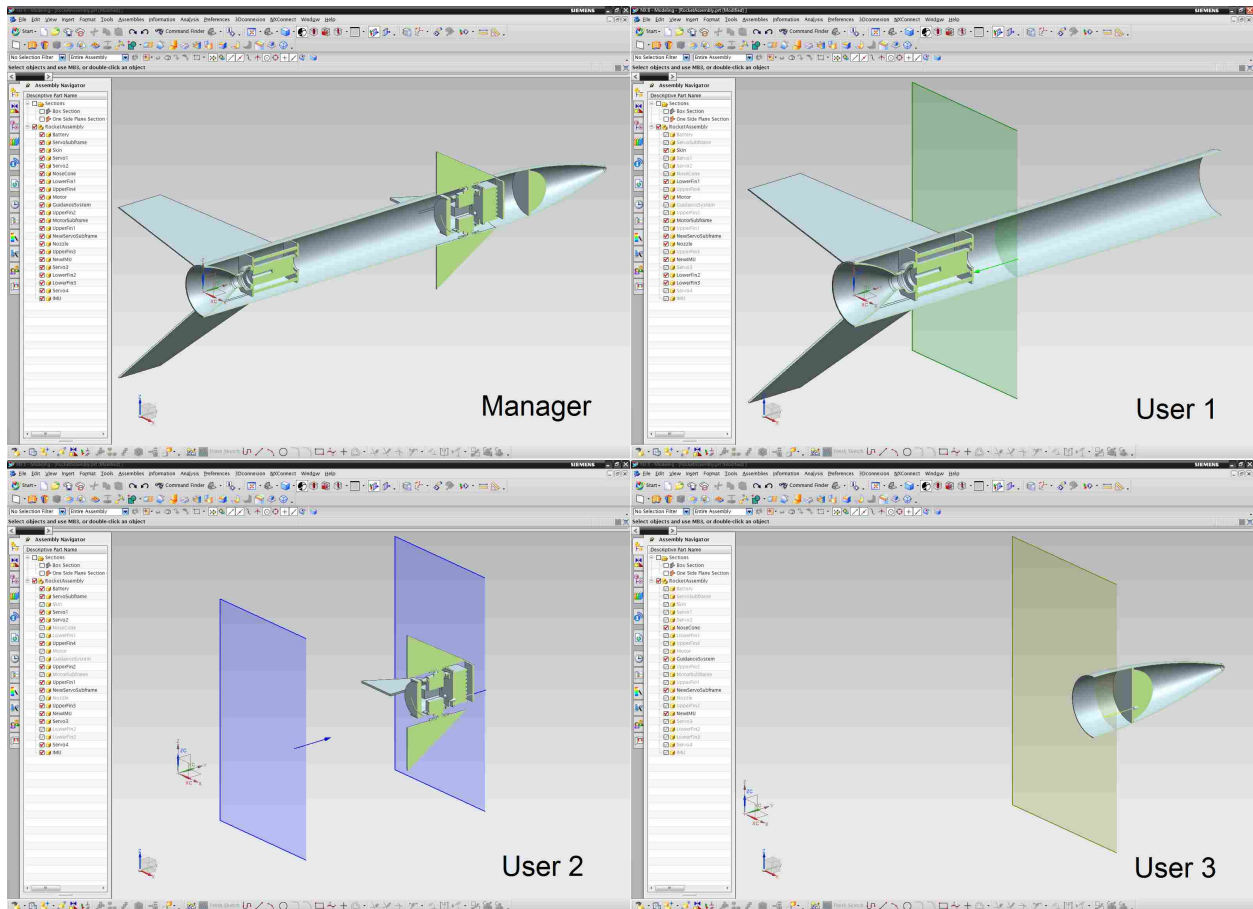


Figure 7.9: Four users' views after being assigned spatial constraint restrictions with the “Hide restricted objects” option enabled.

7.2.3 Administering Access Using a Discretionary Access Control Model

The DAC administration method was tested in NXConnect on the rocket model shown in sections 7.2.1 and 7.2.2.

Since this rocket CAD part was originally created by the user called Manager, he was automatically assigned as a manager for that part. And since this model has been prepared for a certain customer who has access to the multi-user CAD system (whose username is ‘Customer’), the Manager wants to limit access to this part to the Customer and those who are involved in the project. So he assigns the default access level of the part to be Private Access. The Manager also assigns User 1, an engineer who is working on the project, access to edit the CAD model, and assigns the Customer access to view the model as shown in Figure 7.10.

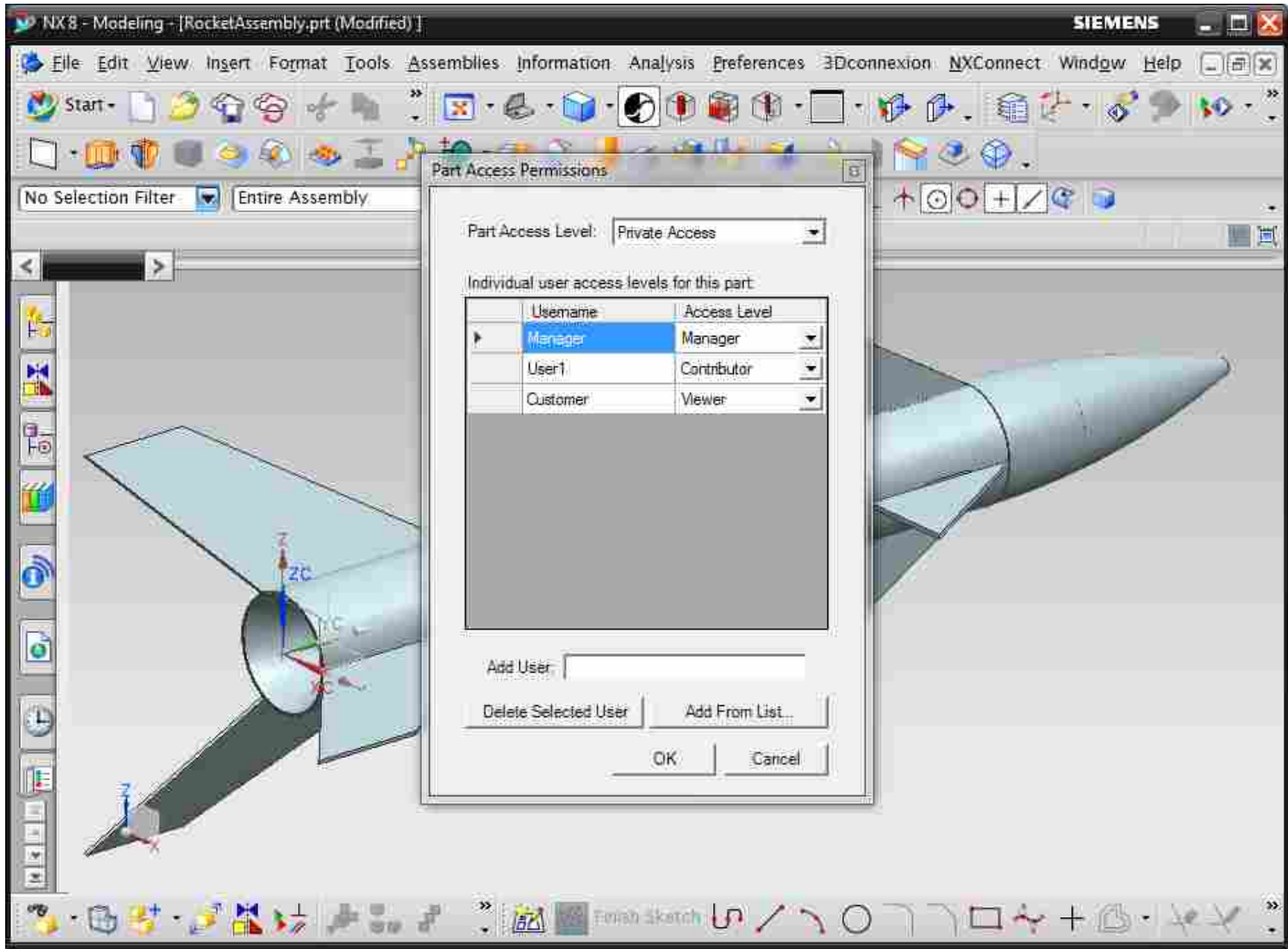


Figure 7.10: The manager of a CAD part assigns other users access to contribute to and view that part using discretionary access control.

While making these assignments the Manager accidentally clicked to change his own access level from Manager to Contributor, but since every part requires at least one manager, the system rejected this change and warned the Manager as shown in Figure 7.11. After completing the access assignments, the Manager click OK to save the changes.

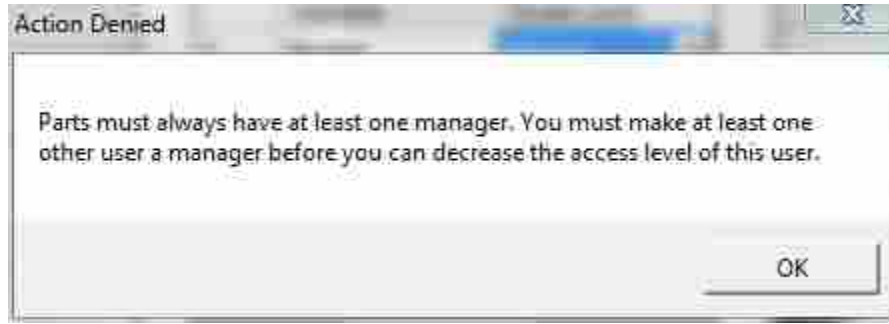


Figure 7.11: An NXConnect error message warns the manager of a CAD part that each part must have at least one Manager.

User 1 opens the model and begins to work on the model. User 1 is under the misunderstanding that he is an administrator for this part, and he attempts to open the spatial constraints administration form. However, since User 1 is assigned to the access level of Contributor for this part, the system rejects his action and displays the error message shown in Figure 7.12.

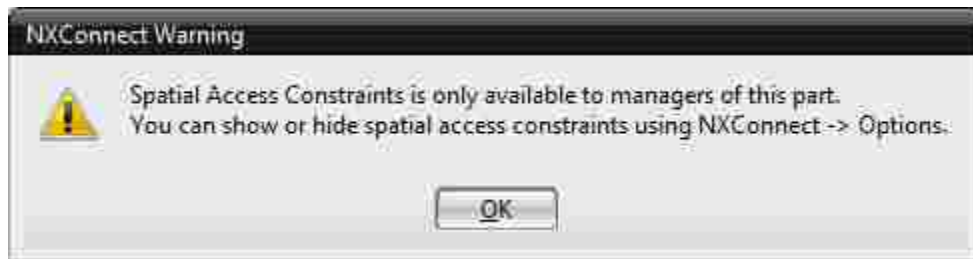


Figure 7.12: A non-manager of a CAD part is blocked from accessing the spatial constraints administration form.

User 1 is surprised that he is not assigned Manager access for this part, so he checks the Part Access form. His view is shown in Figure 7.13. He sees that he is assigned to be a Contributor, and the user called Manager has manage access. User 1 sends an email to the user called Manager to request that his access level for this part be increased.

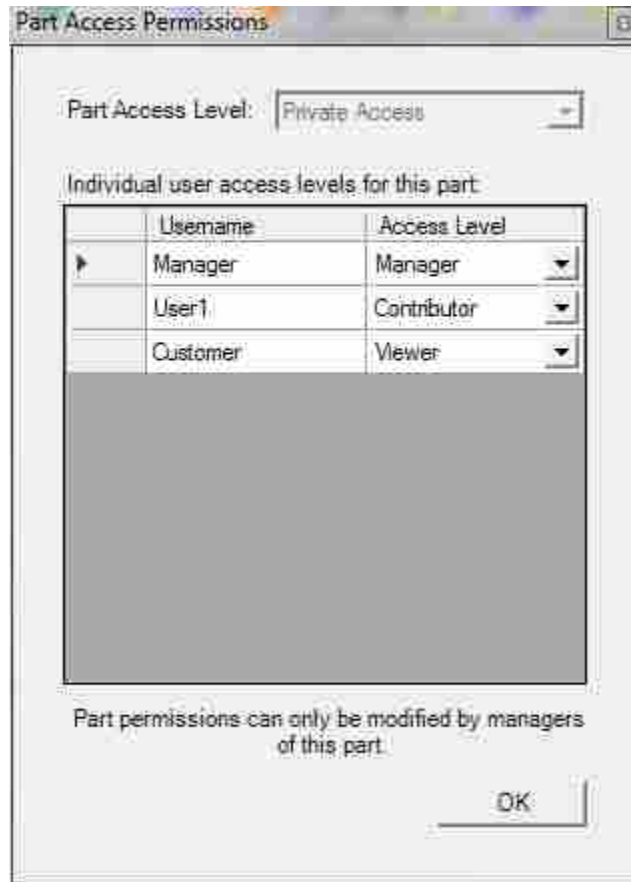


Figure 7.13: A non-manager can view the part access settings but cannot administer them.

The Customer then logs in to NXConnect. He opens the part selection form and begins to type the name of the rocket model. He sees the part available to open as shown in Figure 7.14 and successfully opens it. After opening the part, the Customer tries to delete a feature, but since he is assigned view-only access for this part, his command is blocked as shown in Figure 7.15.

User 2 is not assigned access to this part, and the default access level of this part is set to Private Access, so when User 2 tries to open this part, he sees the “Access Denied” error message shown in Figure 7.16. Since the error message tells him who the manager of the part is, User 2 knows that he can ask the user called Manager for permission to access this part if he needs to.

This example demonstrates the successful administration of access control using a DAC model in multi-user CAD. This method would give engineers and teams flexible control over who has access to perform various actions on their CAD models. This method could also be applied to collaboration with external entities such as suppliers and customers.

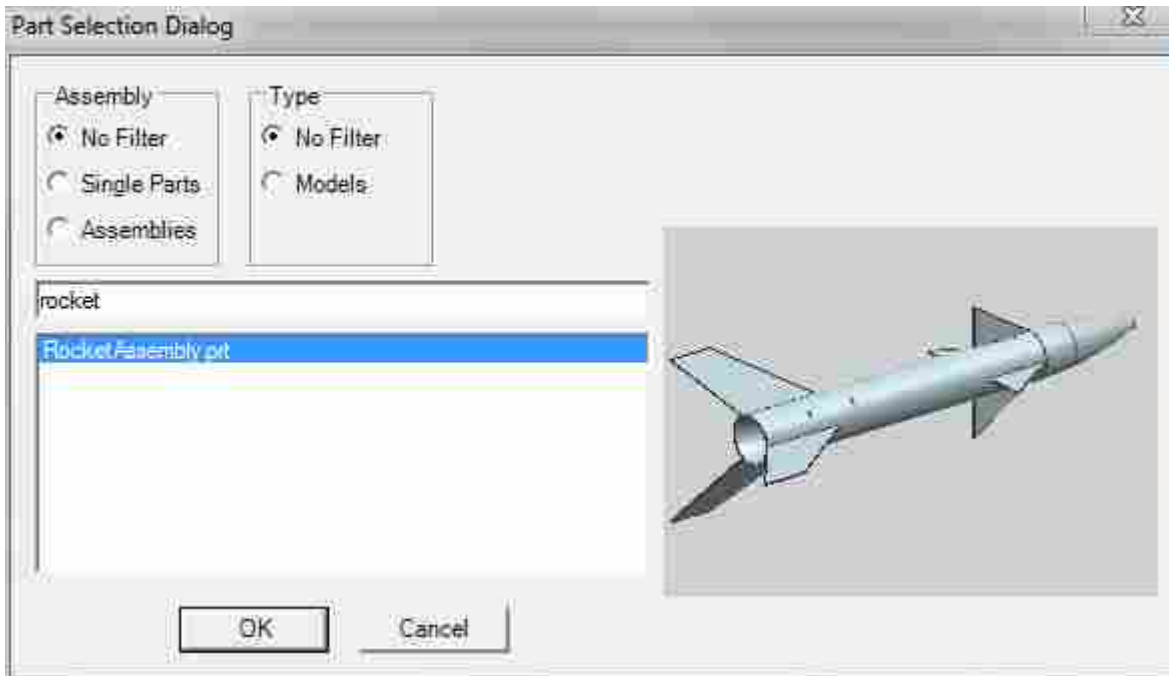


Figure 7.14: A user with view access opens a CAD model.

7.3 Conclusions

This chapter described methods that organizations could use to administer access to sensitive data in multi-user CAD. The following methods were developed and demonstrated in BYU's NXConnect multi-user CAD prototype:

- A method for administering spatial constraints in a CAD part.
- A method for restricting users from viewing a spatial region in a CAD part.
- A method for administering permissions using a Discretionary Access Control model, including restricting users from contributing to (editing) or viewing a CAD part.

These examples showed that a multi-user CAD system can restrict certain user actions such as editing or viewing CAD part or a spatial region within a CAD part while retaining support for multi-user collaboration in a full-featured model. These methods could assist in the practical use of multi-user CAD in engineering industry due to the intense security practices and regulations that affect much of engineering industry. The methods described could be used both within a company and in collaborating with suppliers or customers that have access to the multi-user CAD system.

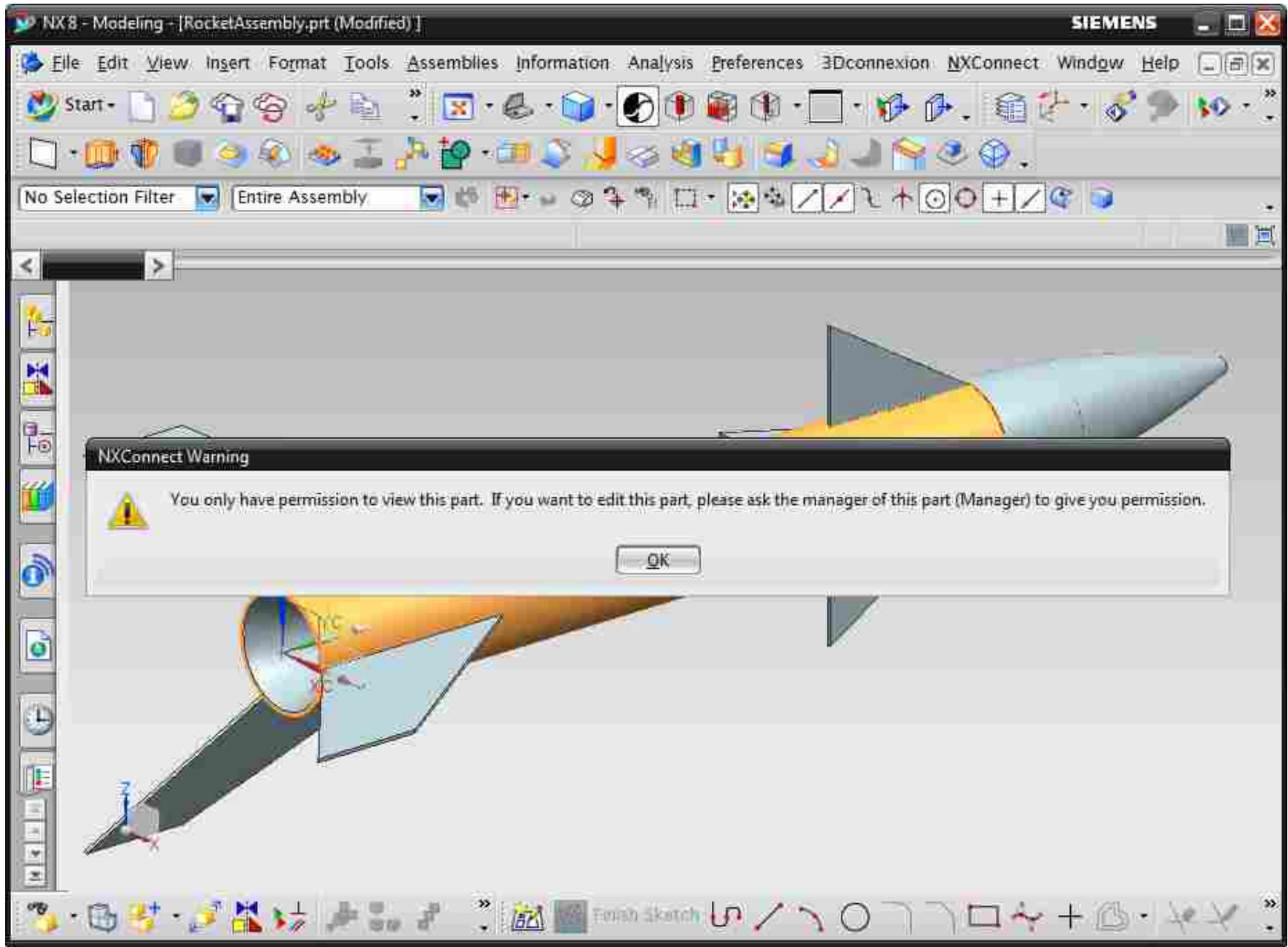


Figure 7.15: A user assigned view-only access to a part is blocked from editing.

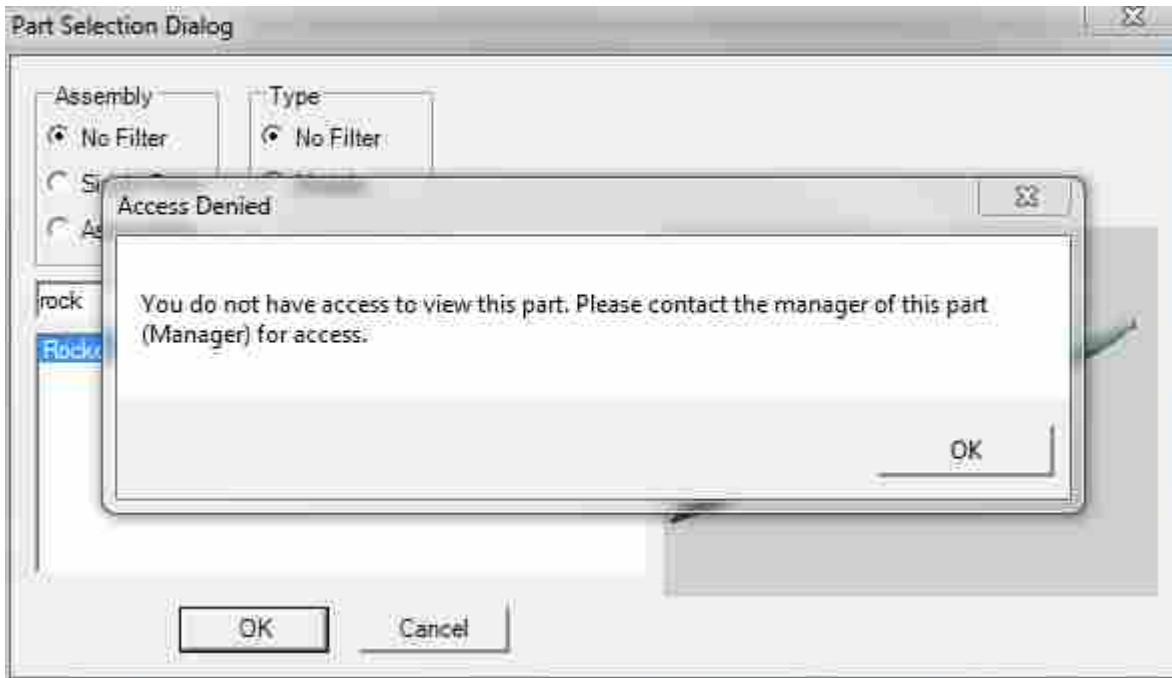


Figure 7.16: A user without view access to a part is blocked from opening it.

7.3.1 Principles

The following principles represent the accomplishment of the objective to develop additional methods for administering and restricting data access in multi-user CAD:

- Spatial constraints can be administered successfully in multi-user CAD.
- Spatial constraints can be used in multi-user CAD to hide regions of the model from specific users while storing the entire model on the client for full model fidelity.
- A discretionary access control model can be successfully administered and enforced in multi-user CAD to give engineers and teams flexible control over their CAD models.

CHAPTER 8. CONCLUSIONS

This dissertation presented human collaboration studies and technical collaboration methods necessary for multi-user CAD to be used effectively in modern engineering industry based on the following principles:

1. Engineers have key insights into what collaboration challenges they are facing and what solutions might help address those challenges.
2. Although multi-user CAD is not yet commonly used in engineering industry, teams of up to forty designers regularly collaborate together virtually, and often simultaneously, in Minecraft to build amazing structures.
3. Local undo in multi-user CAD is very useful, but it allows non-linear undo sequences to occur, which allows conflicts to occur between users.
4. Data access control in multi-user CAD is very important to many engineering companies who want to allow the right collaboration to occur while protecting their sensitive design data.

These principles established the foundation of the human collaboration studies performed and the technical collaboration methods developed.

The results of the two collaboration studies derive the following principles which help us better understand collaboration challenges in engineering industry to understand how multi-user CAD will help address some of those challenges, and help us understand how a multi-user environment affects design collaboration:

1. This study found that collaboration in technical product development involves 50% or more of available work time per day. Multi-user tools promote synchronous and thus immediate

collaboration among a team, which for more complex products will improve the efficiency of team collaboration.

2. Communication awareness is a key characteristic for product development effectiveness when decisions must be made by several technical specialists. Multi-user CAD architectures promote awareness through group decision processes.
3. Because they address specific and immediate problems, ad-hoc meetings prove more effective than formal meetings. Multi-user CAD and other communication technologies provide additional opportunities for virtual ad-hoc meetings.
4. Considering existing single-user product development collaborative tools, there is a need for multi-user collaborative engineering applications in industry.
5. Planning and communication are just as or more important in multi-user design as in single-user design.
6. In multi-user design, effective and accessible communication tools may be even more desirable than colocation.
7. In multi-user design, users strongly prefer working at the same time rather than working at different times.
8. Multi-user design tools allow teams to focus on solving problems together.
9. Our observations of multi-user design in Minecraft were generally supportive of the proposed advantages of multi-user design.

The results of the two technical challenges that were studied derive the following principles which represent the accomplishment of the objectives to develop a classification of types of undo conflicts in multi-user CAD and develop a method for detecting those conflicts, and to develop additional methods for administering and restricting data access in multi-user CAD:

11. Undo conflicts in multi-user CAD can be classified into Feature-Self, Feature-Child, and Feature-Parent types, each with four subtypes.

12. Feature dependencies can be queried to detect and block undo conflicts in multi-user CAD.
13. When an undo conflict occurs, the user can be given information about why it happened and whose command conflicted with theirs.
14. Spatial constraints can be administered successfully in multi-user CAD.
15. Spatial constraints can be used in multi-user CAD to hide regions of the model from specific users while storing the entire model on the client for full model fidelity.
16. A discretionary access control model can be successfully administered and enforced in multi-user CAD to give engineers and teams flexible control over their CAD models.

Point 1 shows that engineers spend a great deal of their time collaborating with each other; therefore, whatever collaborative advantages a multi-user design environment may provide are likely to have a significant effect on designers' work. We believe that multi-user CAD will have a positive effect on collaborative efficiencies by increasing awareness of when communication is needed (Point 2) and by providing additional opportunities for virtual ad-hoc meetings instead of less effective formal meetings (Point 3). It will also be valuable for a multi-user CAD environment to streamline collaboration within its own environment as much as possible.

Point 4 discusses the need for multi-user engineering applications in industry. Point 7 shows that users prefer to work together rather than separately when a collaborative environment is available. Point 9 indicates that multi-user CAD will be valuable and helpful in engineering industry for reducing cycle times and catching design mistakes sooner in the design process. Point 8 shows that users working in a multi-user environment can now focus on what's important: Working together as a team to find and create solutions to their design problems.

Points 5 and 6 show that multi-user CAD will not remove the need for planning and communication; but rather, that these skills will make an even greater difference when using multi-user CAD than when using single-user CAD. And the more communication tools can be optimized to complement multi-user CAD, the better.

Points 11 and 12 form the basis of the method by which local undo conflicts can be successfully detected, which is a necessary prerequisite for resolving those conflicts. Point 13 shows how this method and the information Point 11 refers to can be valuable for resolving conflicts.

An implication of Points 14 through 16 is that a multi-user CAD system can provide both enhanced collaboration and a secure context for that collaboration to occur in, which is very important for modern engineering industry. When combined with the methods developed by other researchers as described in Table 2.2, these methods form a strong basis for administering data access in multi-user CAD.

8.1 Summary

In summary, the study of collaboration in engineering industry revealed how much engineers collaborate, how important communication is in engineering collaboration, and provided evidence of the need for multi-user engineering applications in industry. The study of multi-user design in Minecraft revealed the feasibility and desirability of collaborating simultaneously on a multi-user design with several users, the importance of planning in multi-user design, and that a multi-user design environment allows users to focus on what is important: solving design problems together.

A classification of and method for detecting local undo conflicts in multi-user CAD was developed which can block undo conflicts before they occur and provide users with information about the conflict to help them resolve it. The methods developed for administering data access in multi-user CAD provide a way to restrict sensitive information from unauthorized editing or viewing, and to administer those restrictions using a discretionary access control model.

These four research activities helped us understand the human side of multi-user collaboration and address two important technical collaboration challenges so that multi-user CAD can be used more effectively in a modern engineering environment.

8.2 Limitations and Future Work

The study of engineering collaboration was an intended to be a first look at engineering industry collaboration. It was limited in the number and depth of questions, and the broadness of industry representation. A similar study could be conducted that asked deeper questions and try to get even broader representation from industry.

The study of design collaboration in Minecraft was limited, because although Minecraft is a multi-user design environment there are significant differences between it and CAD. Once multi-user CAD applications become commonly available in engineering industry it would be useful to study or perform an experiment on the use of multi-user CAD itself in engineering industry.

The method for detecting local undo conflicts in multi-user CAD prevent only blocks conflicts from occurring; it doesn't help the user resolve conflicts. Methods for resolving undo conflicts would further improve the undo functionality in multi-user CAD.

The prototype developed for administering spatial regions in multi-user CAD was limited to using datum planes for constraint surfaces. Further development and testing of other kinds of surfaces could be valuable, including testing which types of surfaces are useful for restricting access in different types of CAD models.

The method developed for hiding spatial regions of a CAD model stored the entire model on the client. Further research could be performed to develop a way to only send the client the features that are allowed to be viewed. This could be especially difficult due to the many interdependencies that exist between features and objects in a CAD system, even for features in different regions of a model.

REFERENCES

- [1] “International Slide Rule Museum,” 2014. [Online]. Available: <http://sliderulemuseum.com/SR{ }Ephamera.htm> 1
- [2] A. I. Hepworth, K. Tew, N. Thomas, B. Mark, and C. G. Jensen, “Automated Conflict Avoidance in Multi-user CAD,” *Computer-Aided Design & Applications*, vol. 11, no. 2, pp. 141–152, mar 2013. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/16864360.2014.846070> 2, 3
- [3] E. Red, V. Holyoak, C. G. Jensen, F. Marshall, J. Ryskamp, and Y. Xu, “v-CAX: A Research Agenda for Collaborative Computer-Aided Applications,” *Computer-Aided Design & Applications*, vol. 7, no. 3, pp. 387–404, jan 2010. [Online]. Available: <http://www.cadanda.com/CADandA{ }7{ }3{ }387-404.html> 2, 3, 5, 11
- [4] R. Bidarra, E. van den Berg, and W. F. Bronsvort, “A Collaborative Feature Modeling System,” *Journal of Computing and Information Science in Engineering*, vol. 2, no. 3, p. 192, 2002. [Online]. Available: <http://link.aip.org/link/JCISB6/v2/i3/p192/s1{ }& }Agg=doi> 3
- [5] U. Dietrich, U. V. Lukas, and I. Morche, “Cooperative modeling with TOBACO,” in *Proceedings of the TeamCAD97 Workshop on Collaborative Design*, Atlanta, 1997, pp. 115–122. [Online]. Available: <http://scholar.google.com/scholar?hl=en{ }& }btnG=Search{ }& }q=intitle:Cooperative+modeling+with+TOBACO{ }# }0> 3
- [6] S. Jing, F.-z. He, S.-h. Han, X. Cai, and H. Liu, “A method for topological entity correspondence in a replicated collaborative CAD system,” *Computers in Industry*, vol. 60, no. 7, pp. 467–475, 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0166361509000426http://www.sciencedirect.com/science/article/pii/S0166361509000426> 3
- [7] Y.-c. Y. Kao and G. C. I. G. Lin, “Development of a collaborative CAD/CAM system,” *Robotics and Computer Integrated Manufacturing*, vol. 14, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0736584597000148> 3
- [8] P. Mishra, A. Varshney, and A. Kaufman, “Collabcad: A toolkit for integrated synchronous and asynchronous sharing of cad applications,” in *Proceedings TeamCAD: GVVU/NIST Workshop on Collaborative Design*, vol. vi, 1997, pp. 131 – 137. [Online]. Available: <http://chess.cs.umd.edu/gvil/papers/mishra{ }collab.pdf> 3
- [9] T. Nam and D. Wright, “CollIDE: A shared 3D workspace for CAD,” in *Proceedings of 4th International Conference on Networking Entities*, Leeds, 1998, pp. 103–105. [Online]. Available: <http://65.54.113.26/Publication/2391683> 3

- [10] L. Qiang, Y. Zhang, and A. Nee, "A distributive and collaborative concurrent product design system through the WWW/Internet," *The International Journal of Advanced Manufacturing Technology*, vol. 17, no. 5, pp. 315–322, feb 2001. [Online]. Available: <http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s001700170165><http://link.springer.com/article/10.1007/s001700170165> 3
- [11] K. Ramani, A. Agrawal, M. Babu, and C. Hoffmann, "CADDAC: Multi-Client Collaborative Shape Design System with Server-based Geometry Kernel," *Journal of Computing and Information Science in Engineering*, vol. 3, no. 2, p. 170, 2003. [Online]. Available: <http://link.aip.org/link/JCISB6/v3/i2/p170/s1>[Agg=doi](http://link.aip.org/link/JCISB6/v3/i2/p170/s1) 3
- [12] A. Stork and U. Jasnoch, "A collaborative engineering environment," ... of the *TeamCAD97 Workshop on Collaborative Design*, pp. 25–33, 1997. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Collaborative+Engineering+Environment>[#](http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Collaborative+Engineering+Environment)0 3
- [13] A. Stork, U. Lukas, and R. Schultz, "Enhancing a commercial 3D CAD system by CSCW functionality for enabling co-operative modelling via WAN," in *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, 1998. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Enhancing+a+Commercial+3D+CAD+System+by+CSCW+Functionality+for+Enabling+Co-operative+Modelling+via+WAN>[#](http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Enhancing+a+Commercial+3D+CAD+System+by+CSCW+Functionality+for+Enabling+Co-operative+Modelling+via+WAN)0 3
- [14] T. Nysetvold and C. Teng, "Collaboration Tools for Multi-User CAD," *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*, pp. 1–5, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=6580969 3, 56, 75
- [15] A. Hepworth, K. Tew, M. Trent, D. Ricks, C. G. Jensen, and W. E. Red, "Model Consistency and Conflict Resolution With Data Preservation in Multi-User Computer Aided Design," *Journal of Computing and Information Science in Engineering*, vol. 14, no. 2, p. 021008, mar 2014. [Online]. Available: <http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4026553> 3, 7, 93
- [16] R. A. Moncur, C. G. Jensen, C.-C. Teng, and E. Red, "Data Consistency and Conflict Avoidance in a Multi-User CAx Environment," *Computer-Aided Design and Applications*, vol. 10, no. 5, pp. 1–19, sep 2013. [Online]. Available: <http://www.cadanda.com/CADandA{ }10{ }5{ }727-744.html> 3, 7, 8, 31
- [17] A. I. Hepworth, T. Nysetvold, J. Bennett, G. Phelps, and C. G. Jensen, "Scalable Integration of Commercial File Types in Multi-User CAD," *Computer-Aided Design and Applications*, vol. 11, no. 4, pp. 459–467, jul 2014. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/16864360.2014.881190> 3
- [18] Y. Xu, E. Red, and C. Jensen, "A Flexible Context Architecture for a Multi-User GUI," *Computer-Aided Design and Applications*, vol. 8, no. 4, pp. 479–497, jul 2011. [Online]. Available: <http://www.cadanda.com/CADandA{ }8{ }4{ }479-497.html><http://www.cadanda.com/CAD{ }8{ }4{ }479-497.pdf> 3

- [19] “Google Apps for Work,” 2014. [Online]. Available: <https://www.google.com/work/apps/business/products/docs/> 3
- [20] “Office 365 Business,” 2014. [Online]. Available: <https://products.office.com/en-us/business/office-365-business> 3
- [21] “iCloud,” 2014. [Online]. Available: <https://www.icloud.com/> 3
- [22] “Onshape.com.” [Online]. Available: <https://www.onshape.com/> 4
- [23] D. Crichton, “After 3 Years And \$64M In Seed Funding, Onshape Launches The Mother Of All Products,” 2015. [Online]. Available: <http://techcrunch.com/2015/03/09/onshape-launches-mother-of-all-products/> 4
- [24] J. Mings, “The State of Subscription Pricing — Autodesk, SolidWorks, Solid Edge and Onshape,” 2015. [Online]. Available: <http://www.solidsmack.com/cad-design-news/the-state-of-subscription-pricing-autodesk-solidworks-solid-edge-and-onshape/> 4
- [25] L. Wolter, H. Hayka, and R. Stark, “Improving the Usability of Collaboration Methods and Technologies in Engineering,” in *Digital Product and Process Development Systems*, 2013, pp. 73–84. [Online]. Available: <http://link.springer.com/chapter/10.1007/978-3-642-41329-2>{_}9 5, 11, 12
- [26] M. Contero, P. Company, C. Vila, and N. Aleixos, “Product data quality and collaborative engineering,” *IEEE Computer Graphics and Applications*, vol. 22, no. 3, pp. 32–42, 2002. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/MCG.2002.999786> 6, 12, 25
- [27] R. Heller, A. Laurito, K. Johnson, M. Martin, R. Fitzpatrick, and K. Sundin, “Global Teams: Trends, Challenges and Solutions,” Cornell University, New York, Tech. Rep., 2010. [Online]. Available: <http://www.ilr.cornell.edu/cahrs/hrSpectrum/upload/Spring10Meeting{ }VirtualTeamsWhitePapers.pdf> 6
- [28] E. Red, D. French, C. G. Jensen, S. S. Walker, and P. Madsen, “Emerging Design Methods and Tools in Collaborative Product Development,” *Journal of Computing and Information Science in Engineering*, vol. 13, no. 3, p. 031001, apr 2013. [Online]. Available: <http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4023917> 6, 15
- [29] “Minecraft.” [Online]. Available: <https://minecraft.net/> 6, 14
- [30] “WesterosCraft.” [Online]. Available: <http://westeroscraft.com/> 6, 14
- [31] WesterosCraft, “WesterosCraft Timelapses.” [Online]. Available: <http://westeroscraft.com/timelapses> 6, 14, 64
- [32] T. Cheshire, “Want to learn computer-aided design (CAD)? Play Minecraft,” 2012. [Online]. Available: <http://www.wired.co.uk/magazine/archive/2012/11/play/minecrafterd> 7, 14, 16

- [33] S. Jing, F. He, H. Liu, and B. Liao, "Conflict Analysis in Replicated Collaborative Solid Modeling Systems," *Computer Graphics and Virtual Reality (CGVR)*, pp. 175–181, 2006. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.9133> 7
- [34] B. Liao, F. He, and S. Jing, "Replicated collaborative solid modeling and naming problems," in *Conference on Computer Aided Design and Computer Graphics*, 2005, pp. 3–8. [Online]. Available: <http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=1604644> 7
- [35] A. Hepworth, B. DeFigueiredo, D. Shumway, N. Fronk, and C. G. Jensen, "Semantic Conflict Reduction Through Automated Feature Reservation in Multi-User Computer-Aided Design," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, may 2014, pp. 56–63. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6867542> 7
- [36] X. Cai, F. He, S. Jing, and H. Liu, "A consistency and awareness approach to naming merged faces in collaborative solid modeling," *2008 12th International Conference on Computer Supported Cooperative Work in Design*, pp. 803–807, apr 2008. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4537082> 7
- [37] L. Haftl, "Keeping CAD Files From Going Astray," 2007. [Online]. Available: <http://americanmachinist.com/features/keeping-cad-files-going-astray> 7, 28, 30
- [38] "Export Administration Regulations," 2014. [Online]. Available: <http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear> 8, 9, 29, 48
- [39] "The International Traffic in Arms Regulations," 2014. 8, 9, 29, 48
- [40] C.-C. Teng, F. N. Mensah, J. Ekstrom, R. Helps, and C. G. Jensen, "Hierarchical Role-Based Access Control for Multi-User Collaborative CAD Environment," *Computer-Aided Design and Applications*, vol. 4360, no. December, pp. 1–12, 2015. [Online]. Available: <http://www.tandfonline.com/doi/full/10.1080/16864360.2015.1114398> 8, 30
- [41] R. Moncur, "Data Consistency and Conflict Avoidance in a Multi-User CAx Environment," Ph.D. dissertation, Brigham Young University, 2013. [Online]. Available: <http://db.tt/pZItVNLe> 8, 31
- [42] E. Red, F. Marshall, P. Weerakoon, and C. G. Jensen, "Considerations for Multi-User Decomposition of Design Spaces," *Computer-Aided Design and Applications*, vol. 10, no. 5, pp. 803–815, sep 2013. [Online]. Available: <http://www.cadanda.com/CADandA{ }10{ }5{ }803-815.html><http://www.cadanda.com/CAD{ }10{ }5{ }{ }803-815.pdf> 8, 31, 72, 98, 101
- [43] F. Marshall, "Model Decomposition and Constraints to Parametrically Partition Design Space in a Collaborative CAx Environment," Ph.D. dissertation, Brigham Young University, 2012. [Online]. Available: <http://scholar.google.com/scholar?hl=en{ }&btnG=Search{ }&q=intitle:Model+Decomposition+and+Constraints+To+Parametrically+Partition+Design+Space+In+a+Collaborative+CAx+Environment{ }#> 0 8, 31

- [44] T. Berlage and A. Genau, “From Undo to Multi-User Applications,” *Human Computer Interaction*, 1993. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-57312-7_70 8, 22, 90
- [45] Y. Cheng, F. He, B. Xu, S. Han, X. Cai, and Y. Chen, “A multi-user selective undo/redo approach for collaborative CAD systems,” *Journal of Computational Design and Engineering*, vol. 1, no. 2, pp. 103–115, 2014. [Online]. Available: <http://cadcam.or.kr/upfiles/2013-08-26-EAC-2-4.pdf>http://www.techno-press.com/content/fulltext/j_cde/cde1_2/cde0102004.pdf 9, 28
- [46] R. Absalom and A. Drury, “Collaboration 2.0: Death of the Web Conference (As We Know It),” Ovum, London, Tech. Rep., 2014. 12
- [47] P. Müller, F. Pasch, R. Drewinski, H. Bedenbender, H. Hayka, and R. Stark, “Study on Collaborative Product Development and Digital Engineering Tools,” in *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, L. Rivest, A. Bouras, and B. Louhichi, Eds. Montreal: Springer Berlin Heidelberg, 2012, pp. 389–399. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-35758-9_35{#} 12, 13
- [48] Time, “Game of Blocks: The Man Who Built Westeros in Minecraft.” [Online]. Available: http://content.time.com/time/video/player/0,32068,2402795628001_0,00.html?iid=tabvidrecirc 14
- [49] “Minecraft: Can you undo something in minecraft?!?” 2011. [Online]. Available: <https://answers.yahoo.com/question/index?qid=20110607110246AAXYhsm> 18
- [50] “UnDo,” 2013. [Online]. Available: <http://mods.curse.com/bukkit-plugins/minecraft/undo> 18
- [51] “CommandBlock Undo+Redo,” 2015. [Online]. Available: <https://www.spigotmc.org/resources/commandblock-undo-redo.12840/> 18
- [52] M. David, “A Few Guidelines for a Good Usage of PLM Software,” in *Product Lifecycle Management for Society*, 1st ed., A. Bernard, L. Rivest, and D. Dutta, Eds. Nantes: Springer Berlin Heidelberg, 2013, pp. 29–38. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-41501-2_4 20
- [53] L. Miller and J. Thomas Jr., “Behavioral issues in the use of interactive systems,” *International Journal of Man-Machine Studies*, 1977. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020737377800023> 20
- [54] “Undo,” 2015. [Online]. Available: <http://en.wikipedia.org/wiki/Undo> 20, 21, 22
- [55] Y. Yang, “Undo support models,” *International Journal of Man-Machine Studies*, vol. 28, no. 5, pp. 457–481, may 1988. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0020737388800567> 20, 21
- [56] G. D. Abowd and A. J. Dix, “Giving undo attention,” *Interacting with Computers*, vol. 4, no. 3, pp. 317–342, dec 1992. [Online]. Available: [http://iwc.oxfordjournals.org/cgi/doi/10.1016/0953-5438\(92\)90021-7](http://iwc.oxfordjournals.org/cgi/doi/10.1016/0953-5438(92)90021-7) 20, 21, 26

- [57] A. Prakash and M. J. Knister, “A framework for undoing actions in collaborative systems,” *ACM Transactions on Computer-Human Interaction*, vol. 1, no. 4, pp. 295–330, dec 1994. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=198425.198427> 21, 26, 27
- [58] D. Chen and C. Sun, “Undoing any operation in collaborative graphics editing systems,” *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pp. 197–206, 2001. [Online]. Available: <http://dl.acm.org/citation.cfm?id=500316> 21, 26
- [59] “Command Pattern,” 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Command{ } pattern](http://en.wikipedia.org/wiki/Command_pattern) 21
- [60] B. Meyer, *Object-oriented software construction*, 2nd ed. ISE Inc., jun 1997. [Online]. Available: <http://www.amazon.com/Object-Oriented-Software-Construction-CD-ROM-Edition/dp/0136291554>[http://en.wikipedia.org/wiki/Object-Oriented{ } Software{ } Construction](http://en.wikipedia.org/wiki/Object-Oriented_Software_Construction) 21
- [61] “Software design pattern,” 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Software{ } design{ } pattern](http://en.wikipedia.org/wiki/Software_design_pattern) 22
- [62] “Memento pattern,” 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Memento{ } pattern](http://en.wikipedia.org/wiki/Memento_pattern) 22
- [63] T. Berlage, “A selective undo mechanism for graphical user interfaces based on command objects,” *ACM Transactions on Computer-Human Interaction* (... , vol. 1, no. 3, pp. 269–294, 1994. [Online]. Available: <http://dl.acm.org/citation.cfm?id=196721> 23
- [64] L. Gao, T. Lu, and N. Gu, “Supporting semantic maintenance of complex Undo operations in replicated Co-AutoCAD environments,” in *2009 13th International Conference on Computer Supported Cooperative Work in Design*, 2009, pp. 84–89. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs{ } all.jsp?arnumber=4968039](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4968039) 25, 28
- [65] A. Prakash and M. Knister, “Undoing actions in collaborative work,” ... *on Computer-supported cooperative work*, no. November, pp. 273–280, 1992. [Online]. Available: <http://dl.acm.org/citation.cfm?id=143527> 26
- [66] I. Koren, A. Guth, and R. Klamma, “Shared Editing on the Web: A Classification of Developer Support Libraries,” *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013. [Online]. Available: <http://eudl.eu/doi/10.4108/icst.collaboratecom.2013.254097> 26
- [67] R. Choudhary and P. Dewan, “A general multi-user undo/redo model,” *Proceedings of the Fourth European Conference ...*, pp. 231–246, 1995. [Online]. Available: [http://link.springer.com/chapter/10.1007/978-94-011-0349-7{ }](http://link.springer.com/chapter/10.1007/978-94-011-0349-7) 15 26
- [68] D. Chen, “Consistency maintenance in real-time collaborative graphics editing systems,” Ph.D. dissertation, Griffith University, 2001. [Online]. Available: <http://dl.acm.org/citation.cfm?id=505152> 26

- [69] H. Liu, F. He, X. Li, and Z. Huang, "A less constraint concurrency control and consistency maintainance in collaborative CAD system," *International Conference on Computer Supported Cooperative Work in Design*, pp. 104–109, apr 2010. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5471994><http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=5471994> 27
- [70] Y. Cheng, X. Cai, F. He, and D. Zhang, "A selective undo/redo method in 3D collaborative modeling environment," *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 102–108, may 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6221804> 28
- [71] Y. Cheng, F. He, S. Jing, and Z. Huang, "An multiuser undo/redo method for replicated collaborative modeling systems," in *13th International Conference on Computer Supported Cooperative Work in Design*, 2009, pp. 185–190. [Online]. Available: <http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=4968056> 28
- [72] "Access Control Cheat Sheet." [Online]. Available: <https://www.owasp.org/index.php/Access{ }Control{ }Cheat{ }Sheet> 29
- [73] "Access control," 2016. [Online]. Available: <https://en.wikipedia.org/wiki/Access{ }control> 29
- [74] L. Ding, D. Davies, and C. McMahon, "The integration of lightweight representation and annotation for collaborative design representation," *Research in Engineering Design*, vol. 20, no. 3, pp. 185–200, aug 2009. [Online]. Available: <http://link.springer.com/10.1007/s00163-009-0077-2><http://link.springer.com/article/10.1007/s00163-009-0077-2> 30
- [75] A. Saaksvuori and A. Immonen, *Product lifecycle management*. Springer, 2008. [Online]. Available: <http://link.springer.com/10.1007/978-0-85729-546-0{ }1> 30
- [76] "Google Drive: Change your sharing settings." [Online]. Available: <https://support.google.com/drive/answer/2494886?hl=en> 30
- [77] C. Cera, "Secure and Conceptual Collaborative Computer-Aided Design," Ph.D. dissertation, Drexel University, 2004. [Online]. Available: <http://www.cera.us/cera{ }msthesis/cera{ }msthesis.pdf> 30
- [78] C. D. Cera, I. Braude, I. Comer, T. Kim, J. Han, and W. C. Regli, "Hierarchical role-based viewing for secure collaborative CAD," in *Proceedings of DETC'03*. Chicago, Illinois, USA: ASME, 2003, p. 10. [Online]. Available: <http://gicl.cs.drexel.edu/wiki-data/images/0/0e/ASME-DETC2003-CIE-48277.pdf><http://ukpmc.ac.uk/abstract/CIT/577813><http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1585628> 30
- [79] C. Cera, T. Kim, J. Han, and W. Regli, "Role-Based Viewing Envelopes for Information Protection in Collaborative Modeling," Drexel University, Philadelphia, Tech. Rep. April, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448503002033> 30

- [80] C. Cera and W. Regli, "A collaborative 3D environment for authoring design semantics," *Computer Graphics and Applications*, vol. 22, no. September, pp. 43–55, may 2002. [Online]. Available: <http://www.computer.org/portal/web/csdl/doi/10.1109/MCG.2002.999787>[http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=999787](http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=999787)<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=999787> 30
- [81] C. Cera and I. Braude, "Hierarchical role-based viewing for multilevel information security in collaborative CAD," *Journal of Computing and Information Science in Engineering*, vol. 6, no. 1, p. 2, 2006. [Online]. Available: <http://computingengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1400385> 30
- [82] C. D. Cera, T. Kim, I. Braude, J. Han, and W. C. Regli, "Hierarchical role-based viewing for multi-level information security in collaborative CAD," Drexel University, Philadelphia, PA, Tech. Rep., 2004. [Online]. Available: <https://www.cs.drexel.edu/files/ts467/DU-CS-04-01.pdf> 30
- [83] C.-H. Chu, P.-H. Wu, and Y.-C. Hsu, "Multi-agent collaborative 3D design with geometric model at different levels of detail," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 334–347, apr 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0736584508000070> 30
- [84] G. M. Parker, *Cross-Functional Teams: Working with Allies, Enemies, and Other Strangers*. San Fransisco: Jossey Bass, 2003. 36, 65
- [85] D. Putnam, "Team Size Can Be the Key to a Successful Software Project," McLean, VA, 2005. [Online]. Available: [http://www.qsm.com/process{ }improvement{ }01.html](http://www.qsm.com/process/improvement/01.html) 36, 65
- [86] K. Ferrazzi, "Managing Yourself: Getting Virtual Teams Right," *Harvard Business Review*, pp. 120–123, dec 2014. [Online]. Available: <https://hbr.org/2014/12/getting-virtual-teams-right> 41, 46, 48
- [87] A. Malhotra, A. Majchrzak, and B. Rosen, "Leading Virtual Teams," *Academy of Management Perspectives*, vol. 21, no. 1, pp. 60–70, 2007. [Online]. Available: <http://public.kenan-flagler.unc.edu/faculty/malhotra/LeadingVTsAMP.pdf><http://connection.ebscohost.com/c/articles/24286164/leading-virtual-teams> 47
- [88] S. B. Brunnermeier and S. A. Martin, "Interoperability cost analysis of the US automotive supply chain: Final report," Research Triangle Institute, Tech. Rep. 7007, 1999. [Online]. Available: [http://books.google.com/books?hl=en{ }lr={ }id=0IPD3KnvtTgC{ }oi=fnd{ }&pg=PA1{ }&dq=Interoperability+Cost+Analysis+of+the+U+.+S+.+Automotive+Supply+Chain{ }&ots=LLt9ZmTvW- { }&sig=pEpSOW2JoNeS6tzj7GAqxavpXCU](http://books.google.com/books?hl=en&lr=&id=0IPD3KnvtTgC&oi=fnd&pg=PA1&dq=Interoperability+Cost+Analysis+of+the+U+.+S+.+Automotive+Supply+Chain&ots=LLt9ZmTvW- &sig=pEpSOW2JoNeS6tzj7GAqxavpXCU) 48
- [89] D. Levi, *Group Dynamics for Teams*, 5th ed. SAGE Publications, Inc, 2016. 49
- [90] J. A. Ross, "Trust Makes the Team Go 'Round," *Harvard Management Update*, pp. 3–6, jun 2006. [Online]. Available: <https://hbr.org/2008/02/trust-makes-the-team-go-round-1/> 49
- [91] E. Rocco, "Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact," *Proceedings of the SIGCHI conference on Human factors*

- in computing systems - CHI '98*, no. 496, pp. 496–502, 1998. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=274644.274711> 49
- [92] G. R. Jones, *Organizational Theory, Design, and Change*, 7th ed. Prentice Hall, 2012. [Online]. Available: <http://www.gbv.de/dms/ilmenau/toc/667412913.PDF> 49
- [93] “Organizational structure,” 2015. [Online]. Available: https://en.wikipedia.org/wiki/Organizational_structure 49
- [94] W. Li and Z. Qiu, “State-of-the-art technologies and methodologies for collaborative product development systems,” *International Journal of Production Research*, vol. 44, no. 13, pp. 2525–2559, 2006. [Online]. Available: <http://www.informaworld.com/index/747971734.pdf> 51
- [95] Y. Xie and G. Salvendy, “Agent-based features for CAD browsers to foster engineering collaboration over the Internet,” *International Journal of Production Research*, vol. 41, no. 16, pp. 3809–3829, jan 2003. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/0020754031000149185> 51
- [96] N. Gu, M. J. Kim, and M. L. Maher, “Technological advancements in synchronous collaboration: The effect of 3D virtual worlds and tangible user interfaces on architectural design,” *Automation in Construction*, vol. 20, no. 3, pp. 270–278, may 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0926580510001688> 51
- [97] J. MacMillan, E. Entin, and D. Serfaty, “Communication overhead: The hidden cost of team cognition,” *Team cognition: Process and ...*, 2004. [Online]. Available: http://www.aptima.biz/publications/2004_MacMillan_EntinEE_Serfaty.pdf 51
- [98] “Wikia Minecraft Wiki.” [Online]. Available: http://minecraft.wikia.com/wiki/Minecraft_Wiki 64, 76
- [99] J. Finamore, D. J. Foley, F. Lan, L. M. Milan, S. L. Proudfoot, E. B. Rivers, and L. Selfa, “Employment and Educational Characteristics of Scientists and Engineers,” National Science Foundation, Tech. Rep. January, 2013. [Online]. Available: <http://www.nsf.gov/statistics/infbrief/nsf13311/> 65
- [100] S. Chan, W. Ip, and R. C. P. Chung, “Impacts of project team and innovation process on new product development,” *International Journal of Product Lifecycle Management*, vol. 5, no. 2-4, pp. 224–241, 2011. [Online]. Available: <http://inderscience.metapress.com/index/Y41J38666363111R.pdf> 67, 73
- [101] W. Dyer, J. H. Dyer, and W. G. Dyer, *Team Building*, 5th ed. San Fransisco: Jossey Bass, 2013. 67
- [102] A. Usoro, M. W. Sharratt, E. Tsui, and S. Shekhar, “Trust as an antecedent to knowledge sharing in virtual communities of practice,” *Knowledge Management Research & Practice*, vol. 5, no. 3, pp. 199–212, aug 2007. [Online]. Available: <http://www.palgrave-journals.com/doi/10.1057/palgrave.kmrp.8500143> 68

- [103] D. Bailey and N. Kurland, “A review of telework research: Findings, new directions, and lessons for the study of modern work,” *Journal of Organizational Behavior*, 2002. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/job.144/abstract> 68
- [104] C. D. Cooper and N. B. Kurland, “Telecommuting, professional isolation, and employee development in public and private organizations,” *Journal of Organizational Behavior*, vol. 23, no. 4, pp. 511–532, jun 2002. [Online]. Available: <http://doi.wiley.com/10.1002/job.145> 68
- [105] T. Golden, “The role of relationships in understanding telecommuter satisfaction,” *Journal of Organizational Behavior*, vol. 340, no. October 2004, pp. 319–340, 2006. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/job.369/full> 68
- [106] S. Raghuram, B. Wiesenfeld, and R. Garud, “Technology enabled work: The role of self-efficacy in determining telecommuter adjustment and structuring behavior,” *Journal of Vocational Behavior*, vol. 63, no. 2, pp. 180–198, oct 2003. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S000187910300040Xhttp://www.sciencedirect.com/science/article/pii/S000187910300040X> 68
- [107] “Skype.” [Online]. Available: <http://www.skype.com/> 73
- [108] “TeamSpeak.” [Online]. Available: <http://www.teamspeak.com/> 73
- [109] “Gamepedia Minecraft Wiki.” [Online]. Available: http://minecraft.gamepedia.com/Minecraft{}_Wiki 76
- [110] “World of Minecraft Forum.” [Online]. Available: <http://www.worldofminecraft.com/forum> 76
- [111] K. Jehn, “A multimethod examination of the benefits and detriments of intragroup conflict,” *Administrative science quarterly*, vol. 40, no. 2, pp. 256–282, 1995. [Online]. Available: <http://www.jstor.org/stable/10.2307/2393638> 79
- [112] R. Cosier and D. Dalton, “Positive effects of conflict: A field assessment,” *International journal of conflict management*, no. 1, pp. 81–92, 1990. [Online]. Available: <http://www.emeraldinsight.com/journals.htm?articleid=1659827{&}show=abstract> 79

APPENDIX A. TECHNICAL DETAILS OF UNDO CONFLICT DETECTION METHOD

This appendix provides further details on the local undo conflict detection method described in Chapter 6, Section 6.1.2.

A.1 Data Structure and Acronyms

Figure A.1 through A.3 show the various data required to support this undo conflict detection method. In addition, an operation number is stored on each client that increments with each command applied to the model. This information will be referred to in the discussion below.

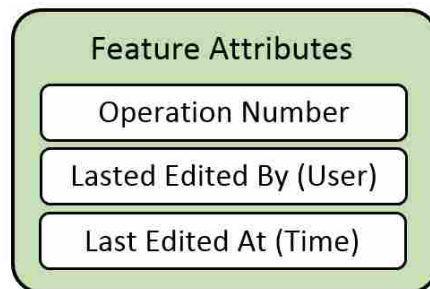


Figure A.1: Data stored as attributes of CAD features.

Also note the following acronyms:

- FS: feature state
- ON: operation number
- MD: metadata, meaning “Last Edited By (User)” and “Last Edited At (Time)”

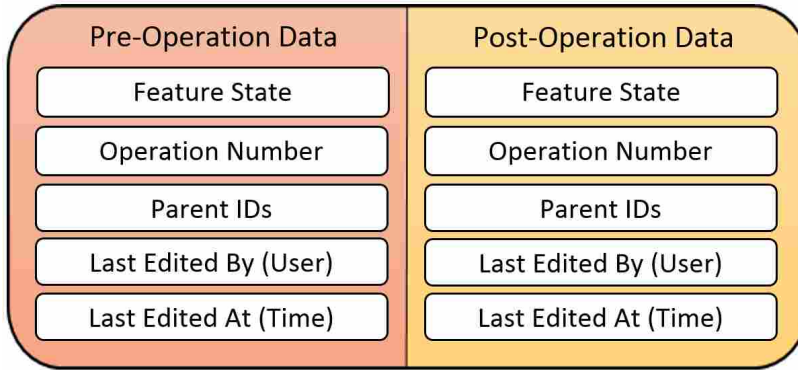


Figure A.2: Data stored in a command in the undo list.

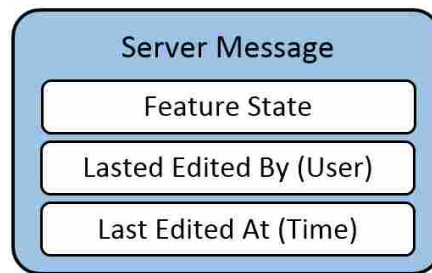


Figure A.3: Data sent to other users.

A.2 Method Details

Section 6.1.2 described a high-level method for checking for undo conflicts. Here is the same method in more detail:

1. Immediately *before* the user performs a *standard* command, set certain pre-operation data.
2. Immediately *after* the user performs a *standard* command, set certain post-operation data.
3. Before allowing undo to execute, check for conflicts.
4. If any conflicts are detected, block the undo, notify the user, and tell the user why the conflict occurred.
5. Immediately after the user successfully performs an undo command, set the feature ON and certain metadata.

Each of these steps are explained in detail in the following sections.

A.2.1 Step 1: Before the User Performs a Standard Command

Immediately *before* the user performs a *standard* command, set the following pre-operation data:

- Pre-operation FS = feature FS
- Pre-operation ON = feature ON
- Pre-operation parents = feature parents
- Pre-operation MD = feature MD

A.2.2 Step 2: After the User Performs a Standard Command

Immediately *after* the user performs a *standard* command, set the following post-operation data:

- Post-operation FS = feature FS
- Post-operation ON and feature ON = new ON
- Post-operation parents = feature parents
- Post-operation MD and feature MD = new metadata

A.2.3 Step 3: Check for Conflicts

Next check for conflicts. Note that in the following discussion, each step only narrows down to two conflict subtypes (e.g. “Type A.1 or Type A.3”). In each case, narrowing down to the exact subtype depends on the intent of the local user’s undo: Refer to Figures 6.3, 6.5, and 6.7 for this information.

Feature-Self (Type A) Conflicts

First check for Feature-Self (Type A) conflicts:

- Has the feature already been deleted? If so, it must have been deleted by someone else, and a Type A.1 or A.3 conflict has been detected.
- Has the current feature been recently edited (compare feature ON to Post-Operation Data ON)? If so, a Type A.2 or A.4 conflict has been detected.

Feature-Child (Type B) Conflicts

Next, check for Feature-Child (Type B) conflicts:

- Have any new child features been recently created (compare feature children to Post-Operation Data children)? If so, a Type B.1 or B.3 conflict has been detected.
- Have any child features been recently edited (compare child feature ONs to Post-Operation Data ON)? If so, a Type B.2 or B.4 conflict has been detected.

Feature-Parent (Type B) Conflicts

Finally, check for Feature-Parent (Type C) conflicts:

- Have any parent features been recently deleted (check if Pre-Operation Data parents still exist)? If so, a Type C.1 or C.3 conflict has been detected.
- Have any of the parent features been recently edited (compare parent feature ONs to Post-Operation Data ON)? If so, a Type C.2 or C.4 conflict has been detected.

A.2.4 Step 4: Handle Detected Conflicts

If any conflicts are detected, take action according to the suggestions in Tables 6.2 and 6.3, including blocking the undo and notifying the user of what happened.

A.2.5 Step 5: After a Successful Undo

Immediately *after* the user successfully performs an *undo* command, set the following data:

- Feature ON = pre-operation ON

- Feature MD = pre-operation MD

A.2.6 Checking for Redo Conflicts

Redo conflicts can be detected by slightly modifying the method used for undo: In Steps 1 and 2, simply swap checking against the pre-operation data or post-operation data. And instead of Step 5, immediately *after* the user successfully performs a *redo* command, set the following data:

- Post-operation MD and feature MD = new metadata
- Post-operation ON and feature ON = new ON