

Development of a Design Environment for Integrated Concurrent Engineering in Academia

by

Simon Nolet

B.E. Mechanical Engineering
Université de Sherbrooke, 1998

Submitted to the Department of Aeronautics and Astronautics in
partial fulfillment of the requirements for the degree of

**MASTER OF ENGINEERING IN AERONAUTICS
AND ASTRONAUTICS**

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

© 2000 Simon Nolet. All Rights Reserved.

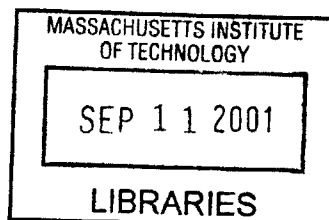
The author hereby grants to MIT permission to reproduce and to
distribute publicly paper and electronic copies of this thesis docu-
ment in whole or in part.

Author
Department of Aeronautics and Astronautics
August 15, 2000

Certified by
Dava J. Newman
Associate Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Certified by
Charles W. Boppe
Senior Lecturer, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by
Nesbitt W. Hagood
Associate Professor, Department of Aeronautics and Astronautics
Chairman, Committee of Graduate Studies



Development of a Design Environment for Integrated Concurrent Engineering in Academia

by

Simon Nolet

Submitted to the Department of Aeronautics and Astronautics on August 15, 2000, in partial fulfillment of the requirements for the degree of Master of Engineering in Aeronautics and Astronautics

Abstract

Among the reforms adopted by the MIT Aeronautics and Astronautics department to converge toward the Conceive, Design, Implement and Operate initiative (CDIO) was the intention to develop a Design Environment for Integrated Concurrent Engineering (DE-ICE) to be used by MIT students and faculty in future design classes. This thesis describes the work done by the DE-ICE team, a group of MIT Master of Engineering students who took up this challenging design project. Two main goals were identified: 1) to develop recommendations for the architecture of the design environment and 2) to discover a key enabler of the system and to design a prototype of the key enabler.

A systems engineering process was followed throughout the project to maintain the traceability between the architecture recommended, the prototype developed and the initial departmental needs. The team benchmarked existing design processes, systems and facilities like Jet Propulsion Laboratory's Project Design Center, Aerospace Corporation's Concept Design Center, TRW's Integrated Concept Development Facility and Caltech's Laboratory for Spacecraft and Mission Design. Interviews with faculty, an analysis of pilot design courses and a survey of students' and faculty's design experience were performed to develop the departmental needs for DE-ICE. Technical requirements were identified and mapped to the needs using a Quality Function Deployment matrix. A use case study and a product matrix were utilized to identify design implementations, from which three architecture variants were developed and evaluated using Pugh's matrix. A final architecture combining elements of the three variants was recommended to the Department.

The On-Line Teaching Assistant (On-Line TA) was identified as the key enabler of DE-ICE. A prototype of the On-Line TA was designed and its functionality is described using Functional Flow Diagrams. Interfaces developed for the demonstration are illustrated. Finally, a system implementation plan for both the recommended architecture and the On-Line TA is presented.

Thesis Supervisor: Dava J. Newman
Title: Associate Professor

Thesis Supervisor: Charles W. Boppe
Title: Senior Lecturer

Table of Contents

Table of Contents	5
List of Figures	9
List of Tables	11
Acknowledgments	13
List of Acronyms	15
1. Introduction.....	19
1.1 Chapter Overview	19
1.2 CDIO: The Context of Engineering Education.....	19
1.3 The Learning Laboratory for Complex Systems	22
1.4 The Microsoft I-Campus Project	25
1.5 The Design Environment for Integrated Concurrent Engineering (DE-ICE) ..	25
1.6 The Team	27
1.7 Thesis Roadmap.....	27
2. Benchmarking Existing Systems and Design Processes.....	29
2.1 Chapter Overview	29
2.2 Literature Review	29
2.2.1 Teaching Engineering Design and Active Learning	29
2.2.2 Facilities and Technologies for Teaching Engineering Design	31
2.2.3 Teaching Distance Courses and Distance Collaboration	35
2.2.4 Integrated Concurrent Engineering in Industry and Government	36
2.3 Site Visits.....	39
2.3.1 TRW.....	40
2.3.2 Aerospace Corporation	44
2.3.3 Jet Propulsion Laboratory (JPL)	48
2.3.4 Boeing (Long Beach).....	53
2.3.5 SAAB.....	54
2.3.6 Caltech Laboratory for Spacecraft and Mission Design (LSMD)	56
2.4 Study of Design Processes Covering the Entire Life Cycle	58
2.5 Chapter Summary	63
3. Education Needs Study For DE-ICE	67
3.1 Chapter Overview	67
3.2 Characterization of the Academic Environment.....	67
3.3 Feedback on Design Pilot Courses	68
3.4 Data Analysis of the DE-ICE Users Experience Survey	72
3.5 Needs Definition and Classification	74
3.6 Constraints	78
3.7 Education Case Study	79
3.8 Chapter Summary	81
4. Design Requirements.....	85
4.1 Chapter Overview	85
4.2 Technical Requirements Selection.....	85
4.3 Requirements Prioritization	87
4.4 Conflicts.....	88

4.5	Use Case Study	89
4.6	Chapter Summary	91
5.	Design Center Architecture Study	93
5.1	Chapter Overview	93
5.2	Design Implementations Study Using a Product Matrix	93
5.3	Architecture Variants	99
5.3.1	Architecture A	99
5.3.2	Architecture B	100
5.3.3	Architecture C	105
5.3.4	Baseline Architecture	106
5.4	Architectures Evaluations	107
5.4.1	Evaluation Through the Product Matrix	107
5.4.2	Evaluation Through Pugh's Matrix	107
5.5	Architecture ABC'	109
5.6	Chapter Summary	113
6.	Prototyping of a DE-ICE System Component	115
6.1	Chapter Overview	115
6.2	Prototype Selection	115
6.3	On-Line TA Use Case Diagram	118
6.4	The On-Line TA Prototype Breakdown	118
6.4.1	Front End for User Interface	119
6.4.2	Front End Authoring Tool for Customizing the On-Line TA	120
6.4.3	Work Manager	121
6.4.4	Project Navigator	122
6.5	Work Manager Functionality	123
6.6	Demonstration	134
6.7	Chapter Summary	135
7.	System Implementation Plan and Recommendations for Future Work	137
7.1	Chapter Overview	137
7.2	Level of Completion of the Prototype	137
7.3	Designer Options	139
7.4	Time and Cost Estimates for Implementing the On-Line TA	140
7.4.1	Estimates for the Option of Creating the On-Line TA at MIT	140
7.4.2	Estimates for the Option of Creating the On-Line TA Externally	144
7.4.3	Selected Option	144
7.5	The Use, Maintenance and Improvement of the System	144
7.6	Learning and Educational Benefits Resulting from DE-ICE	145
7.7	Future Work/Recommendations	146
7.7.1	Future Work To Be Done on the On-Line TA	146
7.7.2	Future Work To Be Done on DE-ICE	146
7.8	Conclusion	147
	References	151
	Additional Sources of Information	153
	Appendix A CDIO Syllabus	155
	Appendix B Design Engineering References	169
	B.1 Reference Books Suggested by Slocum [3]	169

Appendix C DE-ICE Survey	171
C.1 Web Survey (see Farnworth [17])	171
C.2 Results.....	174
Appendix D MIT Operational Modes	187
Appendix E Complete DE-ICE QFD Matrix	193
Appendix F Architecture Variants	195
F.1 Architecture A Physical View (see Farnworth [17])	195
F.2 Architecture A Software/IT Conceptual View (see Farnworth [17])	196
F.3 Architecture A Equipment Block Diagram (see Farnworth [17])	197
F.4 Architecture C Software/IT Conceptual View (see Manka [22])	198
F.5 Baseline Architecture Software/IT Conceptual View (see Farnworth [17])	199
F.6 Baseline Architecture Physical View (see Farnworth [17])	200
Appendix G Prototyping Ideas	201
G.1 The Round Table Virtual Presence System (see Farnworth [17])	201
G.2 The Electronic Writeable Whiteboard (see Farnworth [17])	202
G.3 The Integrated On-Line TA (see Farnworth [17])	203
G.4 Prototype Breakdown (On-Line TA).....	204
Appendix H Demonstration.....	205
H.1 Sequence of Actions	205
H.2 Work Manager Main Interface	206
H.3 Function “7.6 Open Current Work” Interface	207
H.4 Function “7.5 Start New Work” Interface	207
H.5 Function “7.13 Bring Work Home” Interface	208
H.6 Function “7.11 Import External Work” Interface	208
H.7 Function “7.14 Unlock File” Interface	209
H.8 Function “7.7 Update Version in Database” Interface.....	209
H.9 Function “7.8 Submit an Assignment” Interface	210
H.10 Function “7.16 Rename an Entity” Interface.....	210
Appendix I Integrated Information Systems Work Proposal [53]	211

List of Figures

Figure 1.1: Schematic representation of the CDIO syllabus main skills (see Appendix A).....	22
Figure 1.2: The Learning Laboratory for Complex Systems.....	24
Figure 1.3: Description of the Learning Laboratory for Complex Systems.....	24
Figure 1.4: Final schedule of the DE-ICE project.....	28
Figure 2.1: The Georgia Institute of Technology’s VSLCDE (see Mavris [6]).....	32
Figure 2.2: DOME’s distribution of the modeling resources for a particular scenario (see Pahng [8])....	34
Figure 2.3: The decisions affecting the life-cycle cost of designs throughout the duration of the design process (see TRW’s presentation [34]).....	40
Figure 2.4: The ICDF team in action.....	41
Figure 2.5: The ICDF room layout (see TRW’s presentation [34]).....	42
Figure 2.6: The ICDF process (see Heim [12]).....	44
Figure 2.7: The CDC room architecture (see CDC User’s Guide [13]).....	46
Figure 2.8: Interactions among the CDC SST subsystems (see Aguilar [36]).....	48
Figure 2.9: PDC main facility picture (see PDC web site [15]).....	49
Figure 2.10: The Team-X room architecture (see PDC web site [15]).....	50
Figure 2.11: The Team-I room architecture (see PDC web site [15]).....	52
Figure 2.12: Team-I design process scheduling (see Smith [37]).....	52
Figure 2.13: MDO process used at Boeing (Long Beach) for wing design.....	54
Figure 2.14: The SAAB information system architecture (see Farnworth [17]).....	55
Figure 2.15: SAAB structural analysis and modeling process (see Farnworth [17]).....	55
Figure 2.16: Caltech’s Laboratory for Spacecraft and Mission Design (LSMD).....	56
Figure 2.17: NASA program/project life cycle (see NASA Systems Engineering Handbook [19]).....	59
Figure 2.18: Ulrich and Eppinger’s product development process (see Ulrich [20]).....	60
Figure 2.19: Activities performed during the concept development phase (see Ulrich [20]).....	61
Figure 2.20: Process used in 16.89 Space Systems Engineering during Spring 2000.....	62
Figure 2.21: Comparison of the design processes.....	63
Figure 2.22: Comparison between old and new conceptual design processes (see Oberto [38]).....	64
Figure 2.23: ICE process used in LSMD for conceptual design (see Sercel [39]).....	65
Figure 3.1: Design process suggested in 16.00.....	71
Figure 3.2: DE-ICE user needs.....	75
Figure 4.1: DE-ICE QFD matrix.....	86
Figure 4.2: DE-ICE use case diagram.....	90
Figure 4.3: Design use case diagram.....	91
Figure 5.1: DE-ICE product matrix 1 of 4.....	95
Figure 5.2: DE-ICE product matrix 2 of 4.....	96
Figure 5.3: DE-ICE product matrix 3 of 4.....	97
Figure 5.4: DE-ICE product matrix 4 of 4.....	98
Figure 5.5: Architecture B equipment block diagram.....	101
Figure 5.6: Architecture B physical view.....	103
Figure 5.7: Architecture B software/IT conceptual view.....	105
Figure 5.8: Pugh’s matrix for architecture evaluation.....	108
Figure 5.9: Architecture ABC’ software/IT conceptual view (see Farnworth [17]).....	111
Figure 5.10: Architecture ABC’ equipment block diagram (see Farnworth [17]).....	112
Figure 6.1: Prototype selection matrix.....	117

Figure 6.2: On-Line TA use case diagram	118
Figure 6.3: Work Manager use case diagram	121
Figure 6.4: On-Line TA top level FFD	124
Figure 6.5: Log-in service FFD.....	124
Figure 6.6: The On-Line TA Work Manager interface.....	126
Figure 6.7: Work Manager top level FFD.....	127
Figure 6.8: Work Manager detailed FFD 1 of 5	129
Figure 6.9: Work Manager detailed FFD 2 of 5	131
Figure 6.10: Work Manager detailed FFD 3 of 5 (see Manka [22]).....	132
Figure 6.11: Work Manager detailed FFD 4 of 5	133
Figure 6.12: Work Manager detailed FFD 5 of 5	133
Figure 7.1: Prototype level of completion (see Appendix I).....	138
Figure 7.2: Time estimation for the implementation of DE-ICE.....	141
Figure 7.3: Cost estimation for the implementation of DE-ICE.....	142
Figure 7.4: Proposed schedule for implementing DE-ICE	143

List of Tables

Table 1.1: DE-ICE mission statement	26
Table 2.1: Design functions that are best performed by computers and humans (see Neff [9])	35
Table 2.2: Differences between conceptual and detailed design (see Neff [9])	39
Table 2.3: ICDF key team members and their roles	43
Table 3.1: Survey data summary	82
Table 7.1: Time estimates to create On-Line TA elements at MIT	140

Acknowledgments

I would like to thank my teammates, Alex Manka and Bruce Farnworth, for their very good work and their devotion to the project. I am indebted to my advisors, Charles Boppe and Dava Newman, for their support, their availability and their patience, and to Ed Crawley, the Head of the Department, who got the idea of offering this project to Master of Engineering students. I appreciated a lot the inputs of the staff members, Cory Hallam, Fred Donovan and Donna Qualters. The industrial representatives who gave a lot of their time and who were always ready to help us should be mentioned as well: Joel Sercel (Caltech-JPL), David Mitchell (Microsoft), Steve White (Microsoft), Robert Shishko (JPL), Knut Oxnevad (JPL), Billy Fredriksson (SAAB), Kevin Parsons (TRW), Julie Heim (TRW), Todd Mosher (Aerospace Corporation), Jon Neff (Aerospace Corporation), Bob Liebeck (Boeing) and Sean Wakayama (Boeing). Your support and your feedback were extremely important and appreciated. I should not forget Leah Soffer for her help compiling the results of the survey.

Je tiens également à souligner l'aide financière apportée par le Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Fonds FCAR) tout au long de ma maîtrise au MIT. Cela m'a rendu la vie beaucoup plus facile.

Je voudrais remercier toute ma famille, et surtout mon père, ma mère et mon frère, pour leur support tout au long de mes études. Sans eux, je n'y serais jamais arrivé.

Merci aussi à mon coloc, François, de m'avoir encouragé jusqu'au bout.

I would like to thank specially a new person in my life for her support throughout the writing of this thesis. Xueen, thanks for always have been there whenever I needed you. You are the best!

Finalement, je voudrais dédier ma thèse à mon grand-père qui, un jour, m'a posé sur ses genoux, a pris son encyclopédie, et a été le premier et le meilleur professeur de science que j'ai jamais eu. Grand-papa, on y est arrivé!

List of Acronyms

A/A	Aeronautics & Astronautics
ACS	Attitude Control Subsystem
ADR	Architecture Design Review
AI&T	Assembly, Integration and Test
ANSYS	ANalysis SYStem
ASDL	Aerospace Systems Design Laboratory
ASTRO	Architecting the Search for Terrestrial planets and Related Origins
BWB	Blended-Wing-Body
CAD	Computer-Aided Drafting
CAE	Computer-Aided Engineering
CAES	Center for Advanced Educational Services
Caltech	California Institute of Technology
CAM	Computer-Aided Manufacturing
CASDAT	Conceptual Aerospace Systems Design and Analysis Toolkit
CASE	Computer-Aided Systems Engineering
CATIA	Computer-Aided Three-Dimensional Interactive Application
CD	Compact Disk
CDC	Concept Design Center (Aerospace Corporation)
CDIO	Conceive, Design, Implement and Operate
CDR	Critical Design Review
CDS	Computer Data System
CD-ROM	Compact Disk Read-Only Memory
CFD	Computational Fluid Dynamics
COTS	Commercial Off-The-Shelf
CVV	Collaborative Video Viewer
DE-ICE	Design Environment for Integrated Concurrent Engineering
DeMAid	Design Manager's Aid
DMS	Data Management Subsystem
DMU	Digital Mock-Up
DOE	Design of Experiments
DOMÉ	Distributed Object-based Modeling Environment
DSM	Design Structure Matrix
DVD	Digital Versatile Disc
e-Commerce	Electronic Commerce
EBD	Equipment Block Diagram
EDRC	Engineering Design Research Center
EIP	Enterprise Information Portal
E-mail	Electronic Mail
EPS	Electrical Power Subsystem
ESDI	Electrical Subsystem Design and Integration
EWB	Electronic WhiteBoard
EWW	Electronic Writable Whiteboard
FDR	Final Design Review

FEA	Finite Element Analysis
FEM	Finite Element Method
FFD	Functional Flow Diagram
FMECA	Failure Modes, Effects, and Criticality Analysis
FORTRAN	FORmula TRANslation
FSW	Flight SoftWare
G	Graduate
Georgiatech	Georgia Institute of Technology
GUI	Graphical User Interface
H/W	Hardware
ICDF	Integrated Concept Development Facility (TRW)
ICE	Integrated Concurrent Engineering
I-DEAS	Integrated Design Engineering Analysis Software
IIS	Integrated Information Systems
IMAGE	Intelligent Multidisciplinary Aircraft Generation Environment
IP	Internet Protocol
IPD	Integrated Product Development
IPPD	Integrated Product and Process Development
ISE	Intelligent Synthesis Environment
IT	Information Technology
JPL	Jet Propulsion Laboratory (NASA)
LAN	Local Area Network
LCD	Liquid Crystal Display
LSMD	Laboratory for Spacecraft and Mission Design (Caltech)
LTA	Lighter Than Air
LV	Launch Vehicle
Mac	Macintosh
Mbps	Megabits per second
MCR	Mission Concept Review
MDO	Multidisciplinary Design Optimization
MDR	Mission Definition Review
MEng	Master of Engineering
MIT	Massachusetts Institute of Technology
MRAS	Microsoft Research Annotation Service
MS	Microsoft
N2	N-squared
NASA	National Aeronautics and Space Administration
NASTRAN	NAsa STress ANalysis program
OLTA	On-Line Teaching Assistant
On-Line TA	On-Line Teaching Assistant
ORR	Operational Readiness Review
OS	Operating System
PC	Personal Computer
PDC	Project Design Center (JPL)
PDD	Product Definition Database
PDP	Product Development Process

PDR	Preliminary Design Review
PERT	Program Evaluation and Review Technique
ProE	Pro/Engineer
PROP	Propulsion
PSM	Problem Solving Matrix
QFD	Quality Function Deployment
RA	Research Assistant
RDD	Requirements Driven Development
RTVP	Round Table Virtual Presence
S/W	Software
SEMP	Systems Engineering Management Plan
SMS	Structures and Mechanisms Subsystem
SOAP	Satellite Orbital Analysis Package
SOE	School of Engineering
SPHERES	Synchronized Position Hold Engage Re-orient Experimental Satellites
SQL	Structured Query Language
SST	Space Segment Team
STEP	Standard for The Exchange of Product model data
STK	Satellite Tool Kit
SVGA	Super Video Graphics Adapter
TA	Teaching Assistant
TARR	Trade Analysis and Requirements Review
TCS	Thermal Control Subsystem
TELEP	Microsoft Tele-Presence Presentation system
TR	Technical Requirements
TRW	Thompson-Ramo-Wooldridge
TT&C	Tracking, Telemetry & Command
UG	Undergraduate
UML	Unified Modeling Language
VGA	Video Graphics Adapter
VNC	Virtual Network Computing
VSA	Variational Simulation Analysis
VSLCDE	Virtual Stochastic Life Cycle Design Environment
WingMDO	Wing Multi-Disciplinary Optimization
WWW	World Wide Web

Chapter 1

Introduction

1.1 Chapter Overview

Over the past few years, the MIT Aeronautics and Astronautics (A/A) department has developed a Strategic Planning process to determine the future direction of aerospace engineering education. With input from industry, alumni, and faculty, the strategic plan leads to a series of curricular reforms to alter the way engineering is taught in the A/A department. It embraces the notion that engineering education should be in the context of Conceive, Design, Implement and Operate (the CDIO initiative). Infrastructure modifications leading to the development of a new laboratory for education were proposed.

Among the aspects of the strategic plan was the intention of the A/A department to conceive a Design Environment for Integrated Concurrent Engineering (DE-ICE) to be used by students and faculty in future design classes. The challenge was posed to a group of Master of Engineering (MEng) students in Aerospace Engineering, who accepted the task of designing and creating a prototype of DE-ICE. The project was conducted during the Spring 2000 term. This report gives a full description of each of the succeeding steps of the project. It covers every step from the research of existing systems to the identification of the key design drivers, the architecture of the system and the development of a component of the design environment.

The current chapter provides an overview of the main factors that led to the DE-ICE project: the CDIO reform, the Microsoft I-Campus project and the development of the Learning Laboratory for Complex Systems.

1.2 CDIO: The Context of Engineering Education

During the development of the Strategic Planning process, it was realized that education

must consider the holistic view of the engineering process. According to industry, successful engineers are those who not only understand the fundamentals of engineering science, but combine knowledge of the fundamentals with the following skills:

- Understanding of multi-disciplinary design and the manufacturing process
- Ability to manage, to work in teams, and to communicate efficiently
- Ability to operate systems
- Possession of high ethical standards
- Appreciation of the forces at work in a commercial context

The aerospace industry now conceives, designs, implements, and operates very large and complex engineering systems. The A/A department has expanded its mission to better prepare engineers for success and leadership in the Conception, Design, Implementation and Operation of aerospace and related engineering systems. A new integrated education will be created whose goals are (see Crawley [25]):

1. To educate students to master a deep working knowledge of the technical fundamentals
2. To educate engineers to lead in the creation and operation of new products and systems
3. To educate future researchers to understand the importance and strategic value of their work

The plan to implement the CDIO goals is composed of four strategic components (see the Microsoft Proposal [26]):

1. Reforming curriculum and pedagogy, involving the creation of the CDIO syllabus - a detailed listing of the skills and attributes desirable in a young engineer (see Figure 1.1, and also Appendix A for the entire syllabus)

2. Developing a better understanding of technical learning as the basis for educational assessment and improvement
3. Improving faculty competence, in order to develop a cadre facile in the skills of modern engineering practice
4. Building the enabling infrastructure, the new Learning Laboratory for Complex Systems, with new meeting, working, and learning spaces

The vision of this reform is to provide students with an education that stresses the fundamentals, and is focused on real-world systems and products. This is provided by the development of innovative educational programs and pedagogy, and the creation of new research opportunities having the CDIO of engineering systems and products as their educational context. To complement classroom instruction, there will be more open-ended problems, laboratory design-build experiences and student projects leading to the operation of actual aerospace systems. The exercises will be based on authentic activities to build a rich and deep understanding of the concepts and the system development process. More students will be trained in the arts of architecting and integrating:

Architecting:

- Translating the requirements into a system concept
- Visualizing varying designs

Integrating:

- Combining the many subsystem elements into larger systems
- Designing for proper interaction and interdependence among the subsystem elements

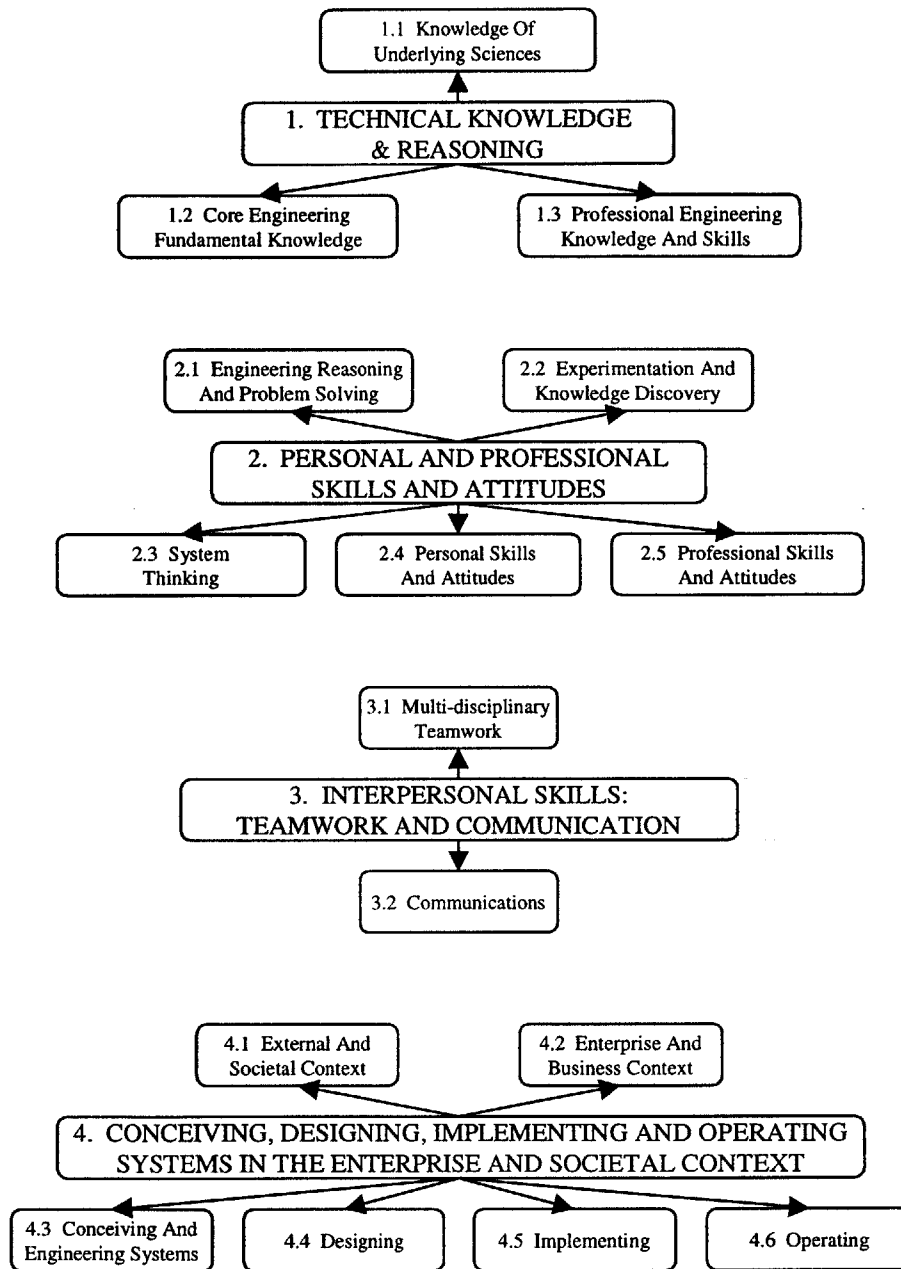


Figure 1.1: Schematic representation of the CDIO syllabus main skills (see Appendix A)

1.3 The Learning Laboratory for Complex Systems

To understand abstract concepts, some learners must manipulate real and concrete objects.

Piaget terms these people “concrete operational” learners (see Wankat [1]). Based on that principle, a key element for delivering successful CDIO projects and enhanced learning is the creation of an environment where multidisciplinary teams can meet and work in real-time (both physically and virtually). The Learning Laboratory for Complex Systems (see Figure 1.2 and Figure 1.3) is one of the main components of the A/A department strategic plan to implement the CDIO initiative. Multidisciplinary teams, composed of both faculty and students in a supportive environment, must have the capability to integrate engineering, manufacturing, business and management factors in their group process. The laboratory, designed to be an inviting and stimulating space that allows students to experiment with many “true-life” variables not always reflected on paper, is organized around the context of CDIO (see section 1.2) and includes:

- A modernized state-of-the-art aerospace engineering library
- A state-of-the-art Design Environment for Integrated Concurrent Engineering
- Workshops, including the “Instrumentation Laboratory”
- A new network
- A new vehicle operations area
- Refurbished wind tunnels
- A totally integrated data communications and computational infrastructure for improved design and teamwork
- Social spaces for students to relax, meet and have fun

The laboratory has the capacity to support 300 people working on multiple projects, classes, and informal activities. It enables students to have access to the computational tools and information resources they need from anywhere in the world. It has the ability to evolve in response to the demands of educational models the Institute follows.

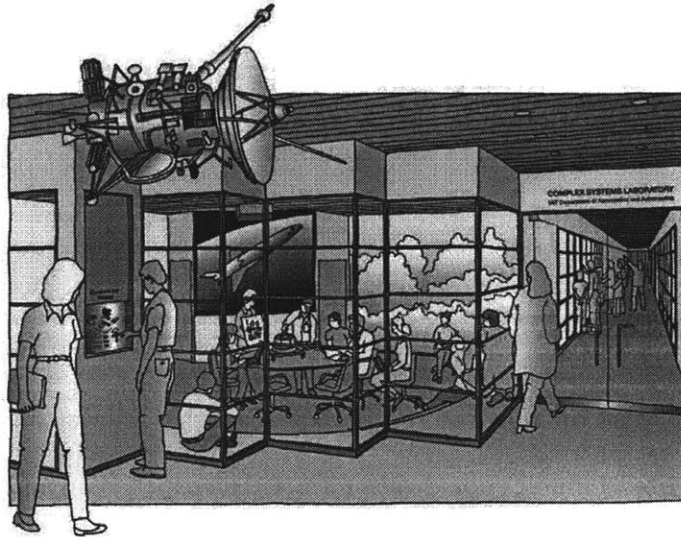


Figure 1.2: The Learning Laboratory for Complex Systems

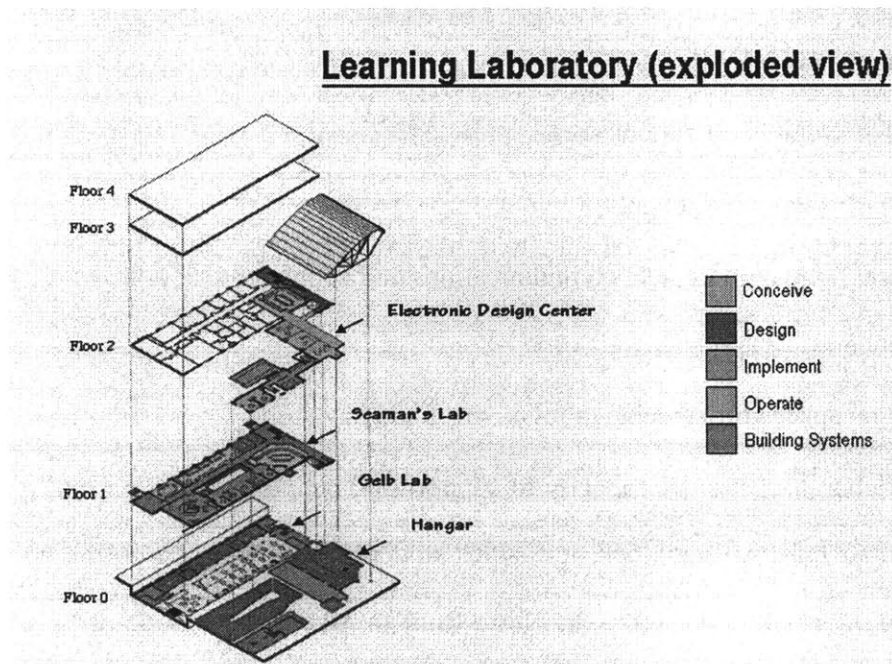


Figure 1.3: Description of the Learning Laboratory for Complex Systems

1.4 The Microsoft I-Campus Project

To succeed in the reform and in the implementation of the Learning Laboratory, the A/A department has decided to build an alliance with world-class companies both in the conducting of the educational experiments and in the evaluation and feedback that will guide the endeavour. Under the MIT/Microsoft alliance named I-Campus, a partnership to forge a new model of university engineering education is proposed. The I-Campus effort in the A/A department is entitled *Active Learning Tools to Enrich Engineering Education* (D. Newman et al., January 2000) and is a joint program with the Civil and Environmental Engineering department. There are three major areas of collaboration under the I-Campus effort in the A/A department (see the Microsoft Proposal [26]):

- Education in distance collaboration for design supported by Information Technology (IT)
- Learning-based education and educational assessment
- Active learning through IT simulations for education

The first phase of the I-Campus effort involved 1) the experimental use of distance collaboration in two graduate design courses and 2) the development of DE-ICE. The design courses were 16.89 Space Systems Engineering and 16.982 Aircraft Systems Design. DE-ICE, the subject of this report, is going to be a major enabler of the Learning Laboratory for Complex Systems. The objectives of the I-Campus effort are 1) to understand the educational benefits (if any) of having students work in a non-located design environment, and 2) to develop the necessary infrastructure and architecture proposed in DE-ICE while collaborating with industry.

1.5 The Design Environment for Integrated Concurrent Engineering (DE-ICE)

The idea of developing DE-ICE as part of a Master of Engineering team-oriented design

project originated in Fall 1999. The DE-ICE project fully supports the CDIO philosophy developed by the A/A department and provides the tools and facilities that will help the students to understand the complex and iterative nature of designing a product, and the global effects of design decisions. This project was seen as a good opportunity for MIT A/A graduate students to participate in the reform program of the A/A department and to gain expertise in the conception of design centers.

Table 1.1: DE-ICE mission statement

Description	To develop an operational framework for a design center to enhance learning in an academic environment.	
Context	DE-ICE will be used in an academic environment to enhance the learning of the process of conception, design, implementation, and operation of complex aerospace systems.	
Goals	To develop recommendations for the architecture of the design environment.	To discover a key enabler of the system and develop a component.
Users	Internal: MIT Students Teaching Assistants Faculty Staff	Possible external users: Guest lecturers Industry representatives
Assumptions	The system should utilize mature technologies.	The system will use existing computer hardware.
Stakeholders	End Users (MIT students) MIT A/A department Microsoft Corporation Industry	

The DE-ICE mission statement is *to develop an operational framework for a design center to enhance learning in an academic environment* (see Table 1.1). It was established by the team members with the help of faculty and staff. The primary design classes in the A/A department using this kind of environment are 16.00 Introduction to Aerospace and Design, 16.423J Aerospace Biomedical and Life Support Engineering, 16.621 Experimen-

tal Projects I, 16.622 Experimental Projects II, 16.684 Experimental Capstone Subject, 16.82 Flight Vehicle Engineering, 16.83 Space Systems Engineering, 16.89 Space Systems Engineering and 16.982 Aircraft Systems Engineering.

1.6 The Team

The three students participating in the project were all involved in the A/A department MEng program. Those students were Bruce Farnworth, Alex Manka and Simon Nolet. The thesis advisors were Professor Dava J. Newman (Microsoft I-Campus Coordinator), Professor Charles W. Boppe (MEng Program Supervisor) and Cory R. A. Hallam (Project Manager of the Learning Laboratory for Complex Systems). Professor Edward F. Crawley (Head of the A/A department, CDIO creator) was also involved in the project.

The following people from industry participated to the project and were present to the reviews: Steve White (Microsoft), Dave Mitchell (Microsoft), Dr. Billy Fredriksson (SAAB), Dr. Robert Shishko (NASA-JPL) and Professor Joel Sercel (Caltech).

1.7 Thesis Roadmap

This thesis is divided in the following sections. The results of benchmarking existing systems and design processes are presented in chapter 2. Then, an education needs study for DE-ICE is achieved in chapter 3, followed by the design requirements establishment in chapter 4. The architecture study of the design center is accomplished in chapter 5. Chapter 6 concerns the work done to select a key enabler of DE-ICE and to design a prototype of that key enabler. Finally, chapter 7 presents a system implementation plan and the recommendations for future work.

The final schedule of the project showing the activities performed during the Spring 2000 semester is illustrated in Figure 1.4.

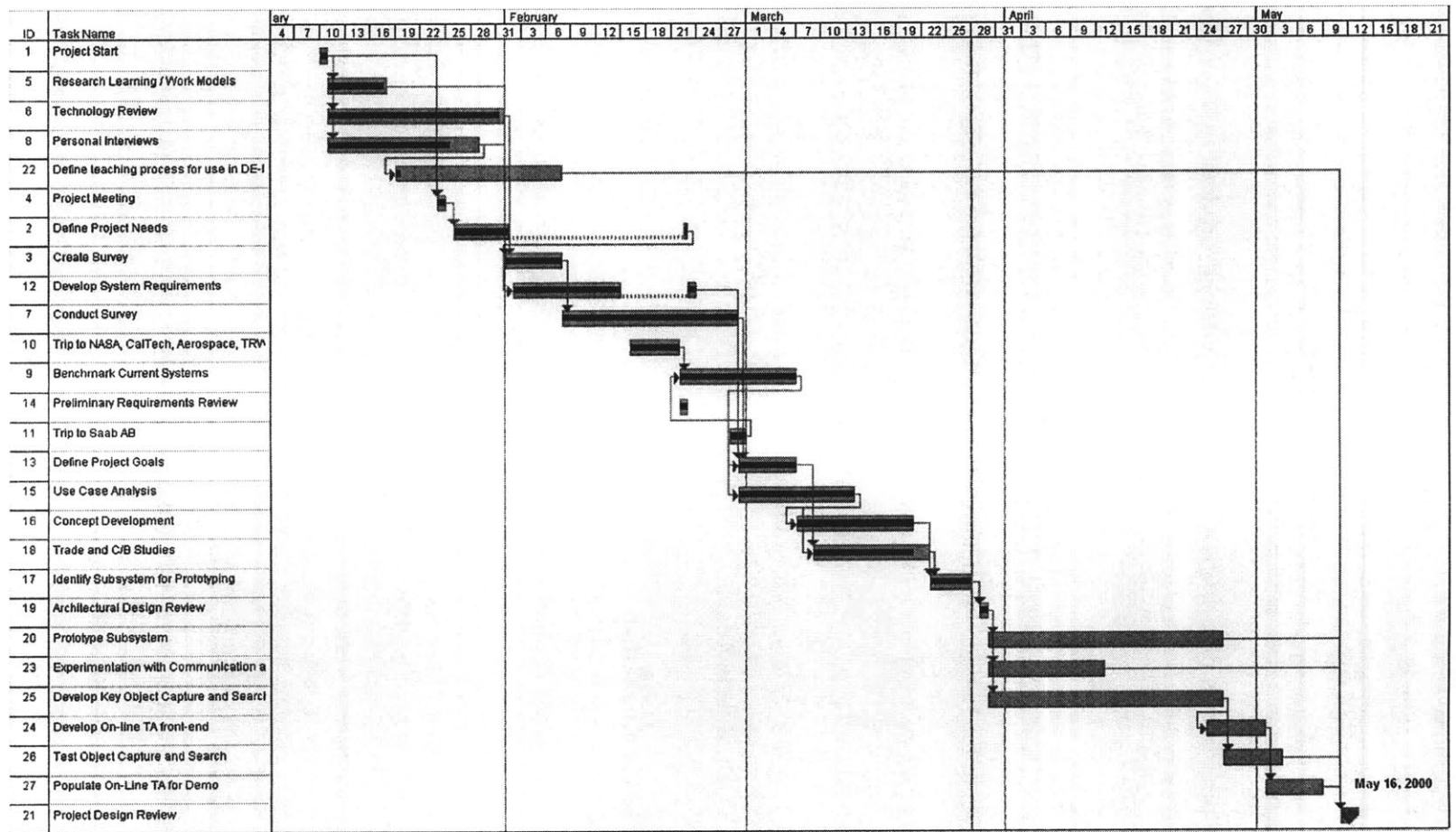


Figure 1.4: Final schedule of the DE-ICE project

Chapter 2

Benchmarking Existing Systems and Design Processes

2.1 Chapter Overview

Recently, the development of new communication and computing technologies has allowed new ways of studying and practicing engineering. It is now possible for people to attend classes from their home or work place, to participate in meetings involving colleagues from all around the world, and even to participate in complex designs from distant locations and keep track of the design evolution in real-time.

The purpose of this chapter is to give an overview of existing research and experience acquired by academia, government and industry both in integrated concurrent design and in distance collaboration. Examples of design centers established in industry and government are given, with specific details on the ones that have been visited for this research effort. An analysis of the various design processes used is made.

2.2 Literature Review

This section provides background literature on integrated concurrent engineering, engineering education, and distance collaboration. Experiences and research achieved by industry, government and academia are highlighted. The four subsections of interest for the development of DE-ICE are: 1) teaching engineering design and active learning, 2) facilities and technologies for teaching engineering design, 3) teaching distance courses and distance collaboration, and 4) integrated concurrent engineering in industry and government.

2.2.1 Teaching Engineering Design and Active Learning

Aerospace engineering education has evolved in the last decade. Now, the objective is to

prepare aerospace engineers to deal with more and more complex problems while cutting cycle time of development and system costs. NASA has already adopted the philosophy of time and cost reduction (see Goldin [2]). The MIT A/A department reform toward the CDIO initiative is a good example of academic evolution focusing on systems engineering, design processes and information technology. Slocum [3] mentioned that “If each engineer understands the structure of the design process and what other members of the team have to do, he or she will be less likely to cause problems that adversely affect the project.” There are numerous examples of engineering departments in academia modifying their curriculum over the last decade to respond to this need from industry.

The Wearable Computer Design course at the Engineering Design Research Center (EDRC, Carnegie Mellon University) is a good example of engineering education evolution. It involves students from four colleges in nine disciplines spread throughout five engineering departments. This class serves as a testbed for learning about multi-disciplinary design teams, geographically-distributed design teams, product design, design processes, design tools, design methodologies and multi-semester courses (see Amon [4]). It begins in the Fall semester with the high-level specifications for a wearable computer, and it ends in the Spring semester with the evaluation of the prototype. Students are assigned to groups and each group designs a subsystem. To improve the design of new products, the design process is recorded through WWW pages, electronic bulletin boards and phase reports. That allows further analysis and evaluation. Lectures and milestone meetings (brainstorming sessions, design reviews and phase report presentations) are archived as well (videotaped). Capturing the history of the project provides feedback and recommendations for use in future courses and for research efforts.

Promoting Active Learning is one of the objectives of the CDIO reform in the A/A department. DE-ICE helps to achieve this objective by giving access to tools and hardware

that enable active learning activities. Richards, Gorman, Scherer and Landel from the University of Virginia provide explanations of active learning through case studies (narrative account of a situation, problem or decision) and instructional modules (open-ended problems); see Richards [5] for more details. Two graduate design classes held at MIT in the A/A department during the Spring 2000 semester are other examples of case studies and instructional modules. Space Systems Engineering (16.89) involved the design of a satellite constellation around Mars to provide navigation and communication to Mars ground assets. Aircraft Systems Engineering (16.982) involved business case studies around the implementation of a very large aircraft. These are the kind of activities to be performed in DE-ICE.

Other activities promoting active learning can be found in Linder [30]. They are classified in four categories: engaging students with subject material, developing interactions among students, developing faculty-student interaction, and incorporating engineering practice. Simulations and games are other variations of active learning activities that can be integrated in engineering classes. The multimedia format has the potential to help students better assimilate knowledge through illustrations and simulations. Engineering cases can involve significant graphical information and numerical data; therefore, DE-ICE facilitates these features.

2.2.2 Facilities and Technologies for Teaching Engineering Design

Through its Aerospace Systems Design Laboratory (ASDL), the Georgia Institute of Technology developed the Virtual Stochastic Life Cycle Design Environment (VSLCDE). The purpose of the VSLCDE is “to facilitate design decision-making over time (at any level of the organization) in the presence of uncertainty, allowing affordable solutions to be reached with adequate confidence” (see Mavris [6]). It was developed after considering the Integrated Product and Process Development (IPPD) concept, which suggests that

downstream manufacturing related issues should be considered in early design phases along with product performance. The five elements composing the VSLCDE (problem formulation/uncertainty modeling, physics-based life cycle modeling, integration, decision making and decision support) are described in Figure 2.1. A design project typically starts with activities being done concurrently in these five elements.

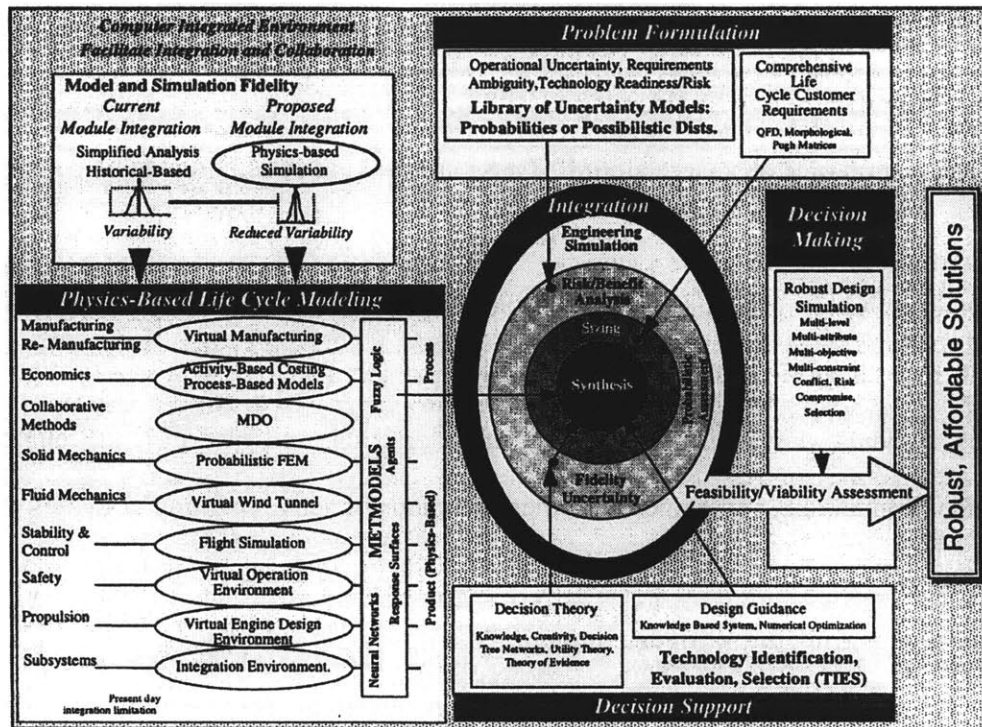


Figure 2.1: The Georgia Institute of Technology's VSLCDE (see Mavris [6])

VSLCDE uses an Internet-enabled design framework called IMAGE (Intelligent Multi-Disciplinary Aircraft Generation Environment). IMAGE is a modular, distributed computing architecture used to assist in design activities. It provides functions for posing design problems, for assembling necessary analyses to tackle them and to execute them. It is an implementation of a complex synergism among information, process and architecture theory. Using IMAGE, a designer can build application-based scenarios to investigate

a variety of engineering systems. The main features and capabilities of IMAGE are (see IMAGE web site [7]):

- A portable Graphical User Interface
- An easy-to-use, graphically driven, object-oriented modeling system
- A structured scheme for integrating analysis
- A plug-n-play mechanism for coupling analysis tools
- A mechanism for running codes on multiple platforms using comprehensive agent technologies
- A method to export a design built in IMAGE to the Web
- A sound environment

At MIT's Computer-Aided Design Laboratory, Center for Innovation in Product Development, the Distributed Object-based Modeling Environment (DOME) is under development. DOME is used for the modeling and evaluation of product design problems in a computer network-oriented design environment using a standard network communication protocol. The DOME framework is intended to (see Pahng [8]):

- Integrate designer-specified mathematical models for multi-disciplinary and multi-objective design problems
- Provide the ability to rapidly construct integrated design problem models to facilitate collaborative design work
- Improve product quality while reducing development time
- Allow specialized engineering applications and design problem models to operate under a common design environment
- Provide a context in which large problems can be modeled through the aggregation

of subproblems.

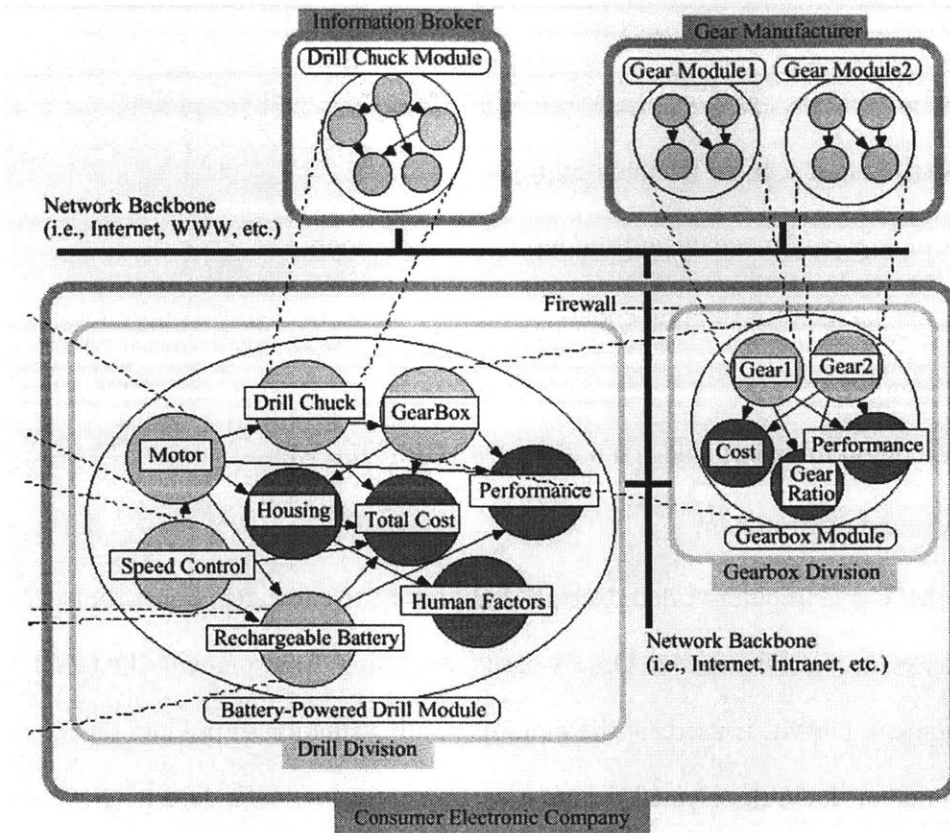


Figure 2.2: DOME's distribution of the modeling resources for a particular scenario (see Pahng [8])

As modern and hi-tech as the new design facilities like the VSLCDE or design frameworks like IMAGE and DOME are, some authors insist on simple and basic hints and techniques that have proven to be very useful over the last decades. Slocum [3] proposes simple but interesting suggestions for design engineers, including reference books (see Appendix B) and useful design materials to do quick solid representations of ideas (cardboard, modeling clay, Styrofoam, foam rubber, paper clips, etc.). These suggestions should be considered when conceiving DE-ICE. A common error is to forget about the educational benefits of simple solid modeling. Often the best results are obtained from a

balanced mix of analysis and experimentation, whether facilitated with modern technology or not. Students should be taught to keep an open mind, and to gather data in the quickest, most efficient, and most accurate method possible. Table 2.1 highlights design functions that are best accomplished by humans and computers. Having a good combination of human and computer tasks supports human strength, compensates for human weaknesses and produces good synergy in design.

Table 2.1: Design functions that are best performed by computers and humans (see Neff [9])

Computers	Humans
<ul style="list-style-type: none"> - Make complex calculations quickly - Organize and store large amounts of informations - Communicate critical design information quickly (when in a network) 	<ul style="list-style-type: none"> - Generate innovative solutions to difficult problems - Apply technical judgement borne of long experience in engineering design

2.2.3 Teaching Distance Courses and Distance Collaboration

With better technologies now available, it is possible for people to attend classes from their home or work place, to participate in meetings involving colleagues from all around the world, and even to participate in complex designs from distant locations. A mode of teaching to distance students that could be eventually used in DE-ICE is through the Internet. A distance teaching experience has been piloted at the University of New Hampshire during the 1998 Spring semester under the Far View Program (see Williamson [10]). Courses were given through the Internet to on- and off-campus students. The software used was *LearnLinc* developed by Interactive Learning International Corporation (ILINC). It produced only real-time audio with *textchat* capability for feedback and questions, and *whiteboard* capability to display and annotate Microsoft PowerPoint slides. The

conclusions and recommendations collected were very similar to the ones made at MIT following two design classes (see Nolet [31]). Learners appreciate the opportunity to interact with practicing engineers and with graduate students from other universities not available otherwise. The unreliability and the limitations of the technology were seen as the major drawbacks. Productive interaction between the lecturer and the students was seen as a real challenge, especially considering the lack of video. Another obstacle is the different pedagogical approach distance teaching requires. As mentioned in Nolet [31], more preparation from faculty is necessary to put the class material in an electronic form. This is especially true the first time a course is given to distance students. This extra preparation work certainly results in higher quality presentations than could be obtained in a traditional lecture (see Williamson [10]).

Interesting results have been collected from a survey made in industry by the University of York (see Brooks [32]). The analysis of the results from industry shows that the most popular personal learning tool is interactive PC and video (far ahead of On-line Internet training and virtual reality). Most organizations responding had an E-mail system but not a conferencing facility, and believed that E-mail was a very effective form of communication.

Some facilities are already in place at MIT to allow distance collaboration and distance teaching. The Center for Advanced Educational Services (CAES) has already conducted a few classes concurrently with other universities like the National University of Singapore and the Nanyang Technological University. The CAES experience shows the high interest generated by the program (see Larson [33]).

2.2.4 Integrated Concurrent Engineering in Industry and Government

The Air Force Research Laboratory in collaboration with Georgia Institute of Technology

developed a Conceptual Aerospace Systems Design and Analysis Toolkit (CASDAT) to provide assessment capability for future military aircraft systems. Interesting lessons learned from implementing and exercising the CASDAT framework are quoted here (see Hale [11]):

- *Modules should be integrated by task or functionality (module capabilities) and not by programs (software tools used).* Often, frameworks are measured by the number of analysis tools that are integrated. More importantly, the capability to perform user-defined tasks should be considered instead.
- *An accurate parameterized geometric modeler is needed.* The model must be parametric (overall geometry defined by easily changeable parameters) so that the configuration can be automatically updated during the use of iterative design procedures. Accuracy is required because the model is translated into various formats depending on the disciplinary analysis tools to be employed.
- *Disciplinary design software file formats are difficult to integrate.* The format and parameterization of disciplinary analysis tools were found to be strikingly disparate. Their integration often required translators that resulted in loss of information or that could function only in a single direction.
- *Revolutionary concepts are difficult to analyze using conceptual tools.* Physics-based programs are typically unavailable, but critically needed to handle the intricacies of revolutionary configurations.

While moving into the next millennium, NASA is expected to face design challenges similar to those faced by industry: reducing cost (total life cycle cost) and new product development cycle time, and increasing safety and product quality. Goldin [2] illustrated the common problem NASA is facing: most of the decisions made early in the development process commit about 90% of the cost when only 10% of the knowledge required is

really understood. And the more cost committed and incurred in the design process, the less available flexibility there is to make the necessary changes. This results in non-optimized design, and is not as effective as it could be (see Goldin [2]). It is crucial for NASA to capture design knowledge at each level of the design process before incurring any significant costs.

To meet these challenges, NASA is developing an Intelligent Synthesis Environment (ISE). ISE is used to create a seamless environment enabling new levels of distributed collaborative teaming. Through ISE, NASA envisions integration of conceptual design, preliminary design, final design, manufacturing, training, maintenance, operations, risk assessment, cost and even product disposal. It is hoped that by bringing disciplines together, NASA will be able to reduce the time to analyze mission design concepts from six months to two weeks. However, only preliminary design is a concern for now. In fact, the gap between specialists doing conceptual design and specialists performing detailed design can be significant (see Table 2.2). The long term goal is to do total end-to-end product life cycle simulation.

As explained in the following section, some companies that have made the shift toward integrated concurrent engineering have teams of specialists dedicated only to conceptual design.

Table 2.2: Differences between conceptual and detailed design (see Neff [9])

Conceptual Design	Detailed Design
Problem is: - Ambiguous - Uncertain - Unframed	Problem is: - Fairly well defined
Requirements are: - “Squishy”	Requirements are: - Fairly stable
Challenge is: - To define the concept - To create a system that hangs together - To consider the system as a whole - To use integrated models to capture ripple effect through the system	Challenge is: - To make the details work - To work on the different parts of the system
Many options at system level	Few options at system level
Process requires a large amount of interaction and coordination	Process can work well with fair amount of individualistic behaviors coordinated by a program office

2.3 Site Visits

At the beginning of the semester, site visits were planned to better understand what industry and governmental organizations are pursuing in terms of design centers and Integrated Concurrent Engineering (ICE). The following companies were visited by MIT DE-ICE team members: TRW, Aerospace Corporation, JPL, Boeing (Long Beach), and SAAB. A description of each design facility is given below. A discussion of the different design processes used in these facilities is also provided.

2.3.1 TRW

The first site visited was the Integrated Concept Development Facility (ICDF) at TRW, Redondo Beach, California. This facility was built to improve design quality and to reduce conceptual design time, where more than 70% of the decisions affecting the life-cycle cost are made (see Figure 2.3). One of the requests was to provide one or more conceptual aerospace vehicle designs (including detailed costs and risk assessments) within a week or less.

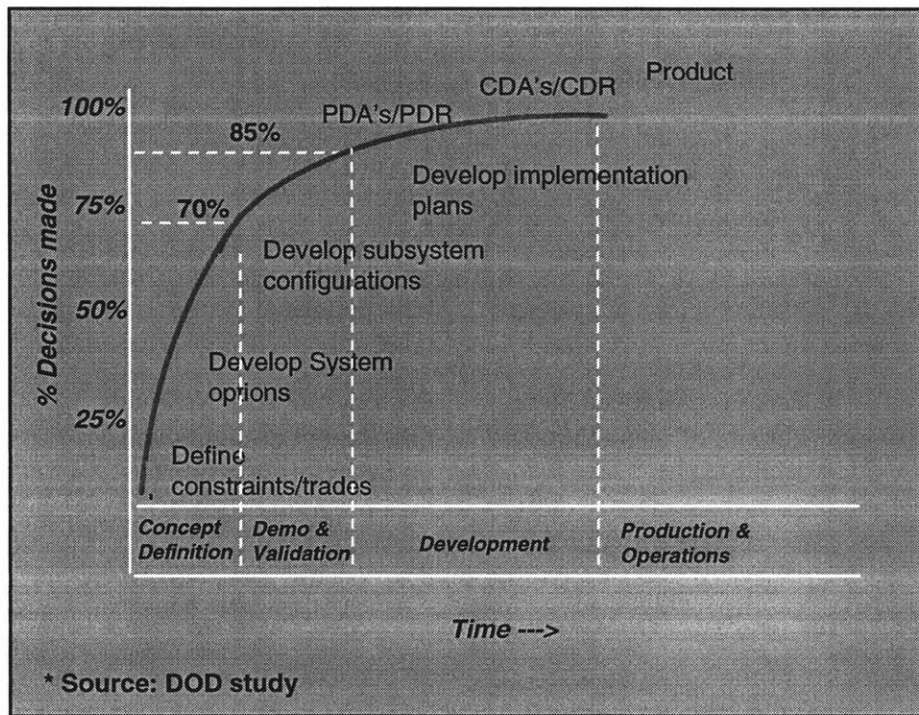


Figure 2.3: The decisions affecting the life-cycle cost of designs throughout the duration of the design process (see TRW's presentation [34])

The studies made in the ICDF vary from feasibility studies (10 years ahead of a project) to preproposal studies (detailed cost estimates). They are executed in real-time by a collocated team divided into subsystem groups. The key elements of the ICDF are (see Heim [12]):

- A structured process and a detailed script
- A dedicated team with well defined responsibilities
- A physical environment
- Automated tools, databases, information exchange media, libraries
- A standard Work Breakdown Structure (WBS) and cost estimating methodology
- A prescribed set of output product templates

The ICDF architecture is illustrated in Figure 2.4 and Figure 2.5. The facility is composed of:

- A main design area to host the design sessions
- A separate conference room where offline meetings can occur without disturbing people located in the main design area
- A storage library where resource documents, vendor specifications and technical manuals can be accessed quickly
- A quiet side room that allows subsystem engineers to think through complex issues, call vendors, or confer with colleagues



Figure 2.4: The ICDF team in action

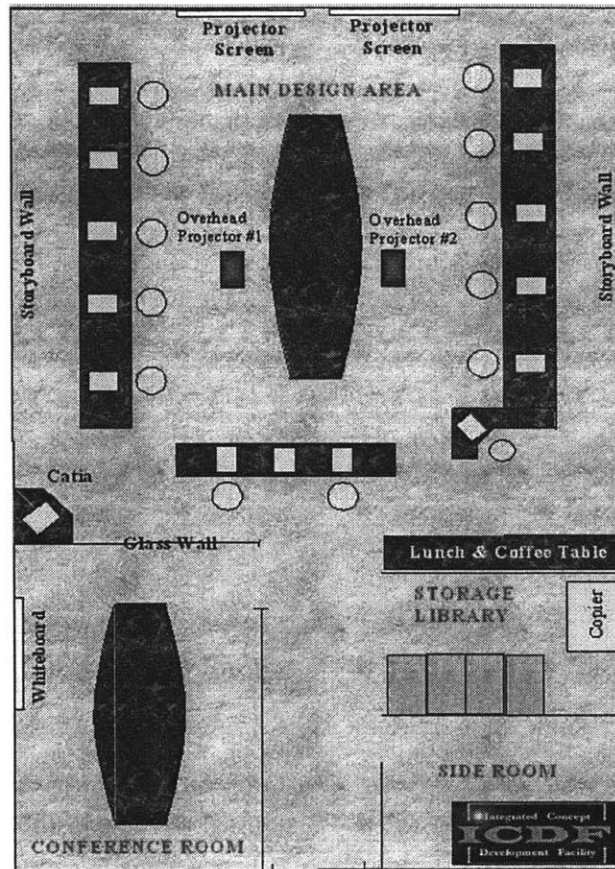


Figure 2.5: The ICDF room layout (see TRW's presentation [34])

The ICDF team is composed of many key members. Table 2.3 describes the design team roles.

The ICDF uses standard products database and tools for communication to ensure that all data is accurate, in a standard format, and is manipulated by a consistent and reliable set of design tools. Tools need to be simple and easy to use so that a backup subsystem engineer wouldn't take too much time to learn how to use them. Excel spreadsheets, Access databases and NetMeeting communications are the most common tools used. The

information and files generated during the design sessions are accessible at any time throughout the company.

Table 2.3: ICDF key team members and their roles

Key Members	Roles
Facilitator	<ul style="list-style-type: none"> - Keeps the design momentum flowing - Ensures that the scripted milestones are met on schedule - Identifies any potential holdups early
Technical Lead	<ul style="list-style-type: none"> - Monitors the technical progress of the design - Verifies that the individual designs of all subsystems tie together - Reviews the design at a system level
Subsystem Engineers	<ul style="list-style-type: none"> - Develop specific portions of a design - Communicate designs to the managers - Own the data that pertains to specific design details
Pricing Representative	<ul style="list-style-type: none"> - Provides costing structure and methodology for competitive bidding
Systems Manager	<ul style="list-style-type: none"> - Maintains system operations - Keeps the data current throughout the design sessions - Coordinates the system updates - Works out any software conflicts - Manages the interfaces between subsystem tools and the ICDF system tool
Database Manager	<ul style="list-style-type: none"> - Ensures that a consistent format is used - Ensures that the data values maintained are current

Figure 2.6 illustrates the process used at TRW to complete a conceptual design study. The process starts with pre-design meetings where the customer defines the customer needs, the requirements, and the very top-level architecture. Political issues are discussed at this point. Then, the design configurations are iterated within the ICDF. This occurs during three sessions of half a day each. For each session, half a day is also required out-

side the ICDF to communicate with colleagues, vendors, the customer, etc. A concept design is the output of the sessions. The process finishes with managers approving the cost and design of a program. Feedback from the team and the customer is solicited to continuously monitor and improve the process.

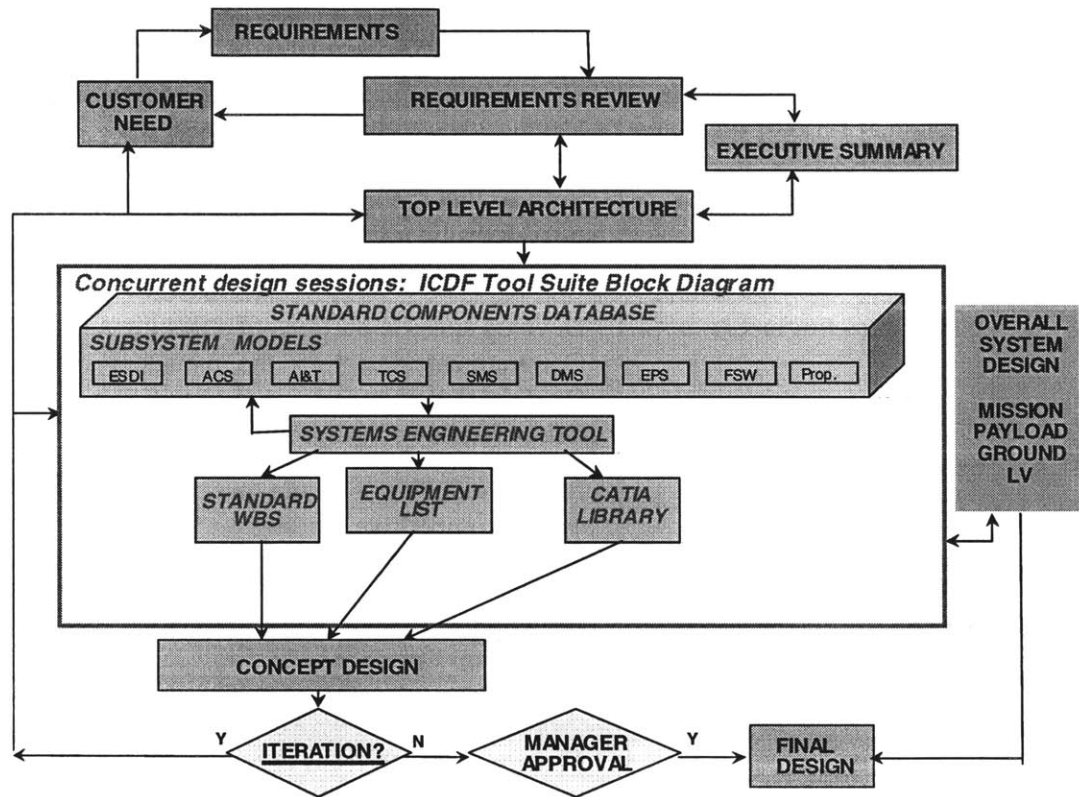


Figure 2.6: The ICDF process (see Heim [12])

2.3.2 Aerospace Corporation

The second site visited was the Concept Design Center (CDC) at the Aerospace Corporation headquarters, El Segundo, California. Like TRW's ICDF, the CDC was conceived to create and evaluate conceptual designs in a synergistic and concurrent manner, recognizing the value of the up-front system engineering process to reduce time and cost to develop new design concepts (see the video produced by Aerospace Corporation [35]).

The CDC is composed of three key elements (see Aguilar [36]):

1. Teams of multi-disciplinary experts
2. Facilities where the customer can interact efficiently with the teams of experts
3. A process for applying innovative design tools to produce quality results quickly

Each team is divided into groups assigned to a subsystem (like communications, thermal, power, etc.). Each group is composed of at least one expert. Systems engineers, whose role it is to capture internal (subsystems to subsystems) and external (specific constraint/requirements to subsystems) system interfaces, are an integral part of the team. The customer is also part of the team to provide feedback on the mission, goals, and the critical decisions made during the study. The CDC is mainly used for conceptual design, technology insertions and trade space exploration. Experience has shown that this collaborative design is best suited to system-level issues (see Neff [9]).

The CDC facility is shown in Figure 2.7. The subsystem groups with the most interaction are located next to each other. Overhead projectors are available to project any monitor focusing on specific design aspects. The cost of the facility is estimated at \$500,000 for the first year (implementation), and then about \$100,000 per year for maintenance. Four people are working half time to maintain the facility and the models used.

The tools used in the CDC are: Microsoft Excel for the environment for nearly all of the modules developed by the specialists, Lotus Notes for communications, and Team Effectiveness Tool (TET) as a way to regularly check on team dynamics and key CDC processes. All of these tools are quite simple to use, and experience shows that the success of a new collaborative conceptual design process is usually inversely proportional to the time and effort required to learn the information system (see Neff [9]). The CDC does not

use an extensive database of parts. The main reasons are the difficulty of implementing and maintaining databases, and the difficulty of convincing people to use them.

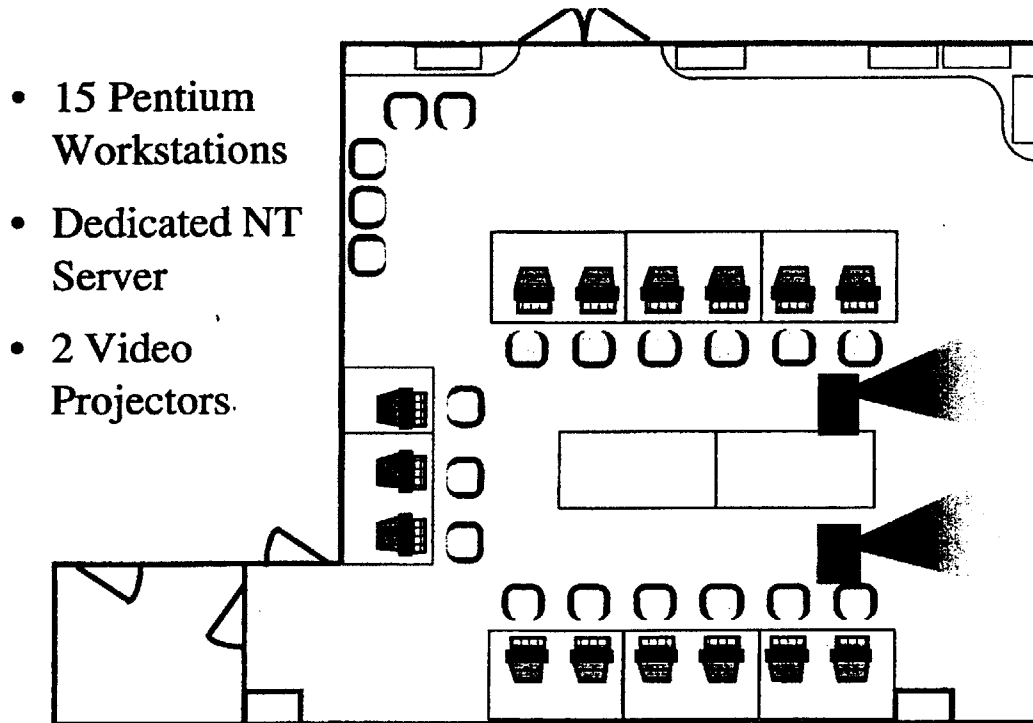


Figure 2.7: The CDC room architecture (see CDC User's Guide [13])

People at Aerospace Corporation do not have a fully integrated software for their design sessions since conceptual design does not deal a lot with numbers. They mentioned the difficulty of development and the fact that by the time the software is running, it is already obsolete.

The main metric used to evaluate the performance of the CDC is the time taken to produce a conceptual design for a spacecraft. Overall reductions of 50% to 75% have been observed. Alternatively, the CDC provided greatly enhanced detail for an equivalent resource expenditure (see Aguilar [36]).

Experience has shown that the human factor is very important in collaborative design. Some people educated in the traditional way might have trouble dealing with a new interactive environment. Team members need to have good human interaction skills as well as the required technical skills in order to create a productive team that is able to handle the collaborative environment.

The CDC process is divided into three major milestones:

1. Study Planning (two to eight weeks prior to the CDC session)

During this phase, the systems engineer works with the customer. Together, they scope the problem and define the statement of work. The team members do independent research and analysis, and develop the software they will use in the CDC session. Political decisions about the design (like the decision of doing a reusable launch vehicle, for example) have to be made prior to the CDC session.

2. CDC Session (three hours sessions, two to four times)

Fifteen experts divide into subsystem teams to take part in the CDC design sessions. During these sessions, the configuration of the project is generated. The real-time design uses linked distributed software models on spreadsheets. All the different subsystem teams work concurrently, and all of the subsystems are linked to each other through a local file server. A change made to one subsystem affects others instantaneously. This aspect allows for a rapid check of the impact of a modification to the overall system. Figure 2.8 shows interactions among the CDC Space Segment Team (SST) subsystems. A lot of these interactions are verbal among the engineers. Subsystem interactions not captured electronically are documented later. The customer is always present at the CDC sessions.

3. Post-Session Activities (up to four weeks after the CDC session)

The on-going support and the project report, a subsystem component level description of the conceptual system, compose the Post Session Activities (see the video produced by Aerospace Corporation [35]). It records all the assumptions made during the design. It is delivered to the customer two weeks after the CDC session period. Risk and cost are emphasized in the report. A cost section provides not only a life cycle cost for the entire system, but also a cost-risk analysis (see Aguilar [36]). The on-going support consists of responding to customer requests for modifications.

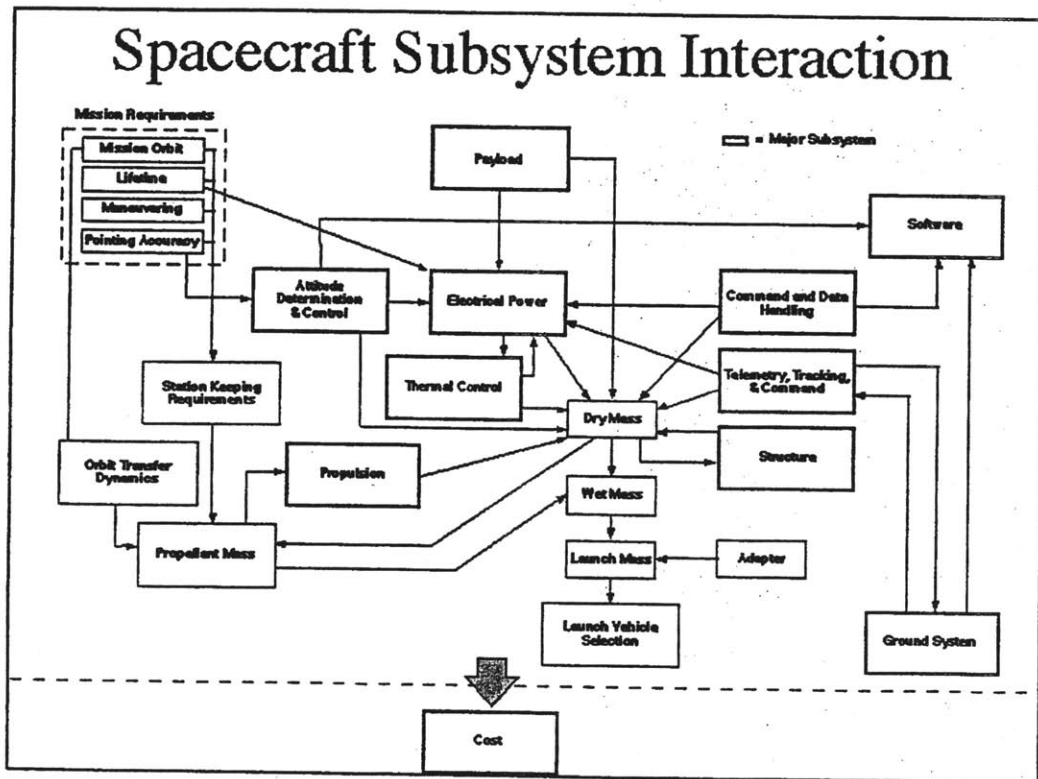


Figure 2.8: Interactions among the CDC SST subsystems (see Aguilar [36])

2.3.3 Jet Propulsion Laboratory (JPL)

The third facility visited was the Project Design Center (PDC) of the Jet Propulsion Labo-

ratory (JPL) located in Pasadena, California. PDC was formed to develop, promote and support the use of concurrent engineering techniques by JPL design teams, especially at the earliest phases of the design. It employs new processes that take advantage of real-time computations to explore the design space with all relevant disciplines present (see Smith [14]). It uses the Concurrent Engineering Methodology, a collection of techniques, lessons learned, rules of thumb, algorithms, and relationships developed for conceptual space system design by Aerospace Corporation. The facility provides similar equipment to the design centers at TRW and Aerospace Corporation: computers, projectors, audio/video conferencing, network connection, etc. (See Figure 2.9). It consists of three separate meeting areas used by different design teams.



Figure 2.9: PDC main facility picture (see PDC web site [15])

Two concurrent engineering teams use the PDC: Team-X for space mission design and Team-I for spacecraft instrument design. Team-X was formed to enable JPL to produce a larger number of space mission proposals with the same amount of resources under the new NASA philosophy *faster, better, cheaper*, adopted in 1992. Cost control in the proposals is now as important as mission performance (see Smith [14]). Team-X's architecture is illustrated in Figure 2.10.

The main integrated tools used by Team-X are distributed Excel spreadsheets linked using the Macintosh utility Publish and Subscribe (utility that allows users to share and update data in models), a few databases of spacecraft components, distributed documentation software, graphics software to produce 3-D images and subsystem design tools for each subsystem. Team-X has a preference for the Macintosh computer platform.

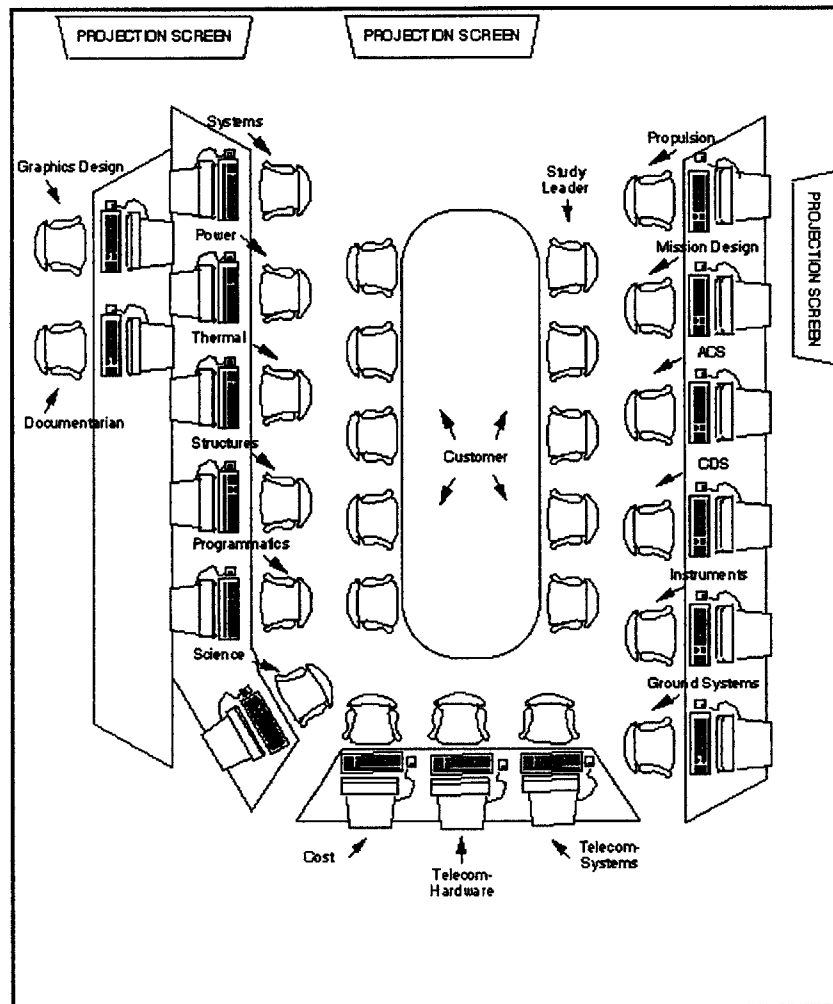


Figure 2.10: The Team-X room architecture (see PDC web site [15])

The design process is similar to the one at TRW and Aerospace Corporation. The customer is present throughout the process. The first step is for the customer to contact the

Team-X leader for preliminary discussions (desired technical results, desired outputs, cost model preference, report form, etc.). After an agreement, the first design session begins with the customer briefing Team-X on the mission and the job requested. Then, the Team-X leader takes control of the session. A scripted procedure is followed. A minimum of two sessions of three hours each separated by a few days is always required to complete even the simplest conceptual design. Complex missions can require up to ten design sessions. Team-X does not search for optimal designs, but for one that meets all mission requirements at acceptable costs and risks. The final report documents the results of the conceptual design. It is drafted in real-time during the design sessions, and is ready one week after the final design session.

Team-I was inaugurated to design space instrument concepts instead of space missions and spacecraft. Space instrument concepts require optical, structural and thermal analyses during the design. These analyses use high-end design tools: computer-aided drafting, engineering and manufacturing tools (CAD/CAE/CAM), STEP (Standard for The Exchange of Product model data) file format, and thermal analysis tools. STEP is a file format used to represent and exchange digital product information. It covers geometry, topology, tolerances, relationships, attributes, assemblies, configuration and more. Most of these tools used by Team-I are Unix-based, while a few of them are PC based. The integrating software is LabView.

To support these more detailed analyses, the concurrent design process was modified. Significantly more off-line work was added prior to and between the concurrent sessions to develop and parametrize CAD models, for example (see Figure 2.12). Once the basic design is captured and parametrized, it can be employed in real-time ICE sessions. This process could be applicable in some way to more detailed design, like preliminary design.

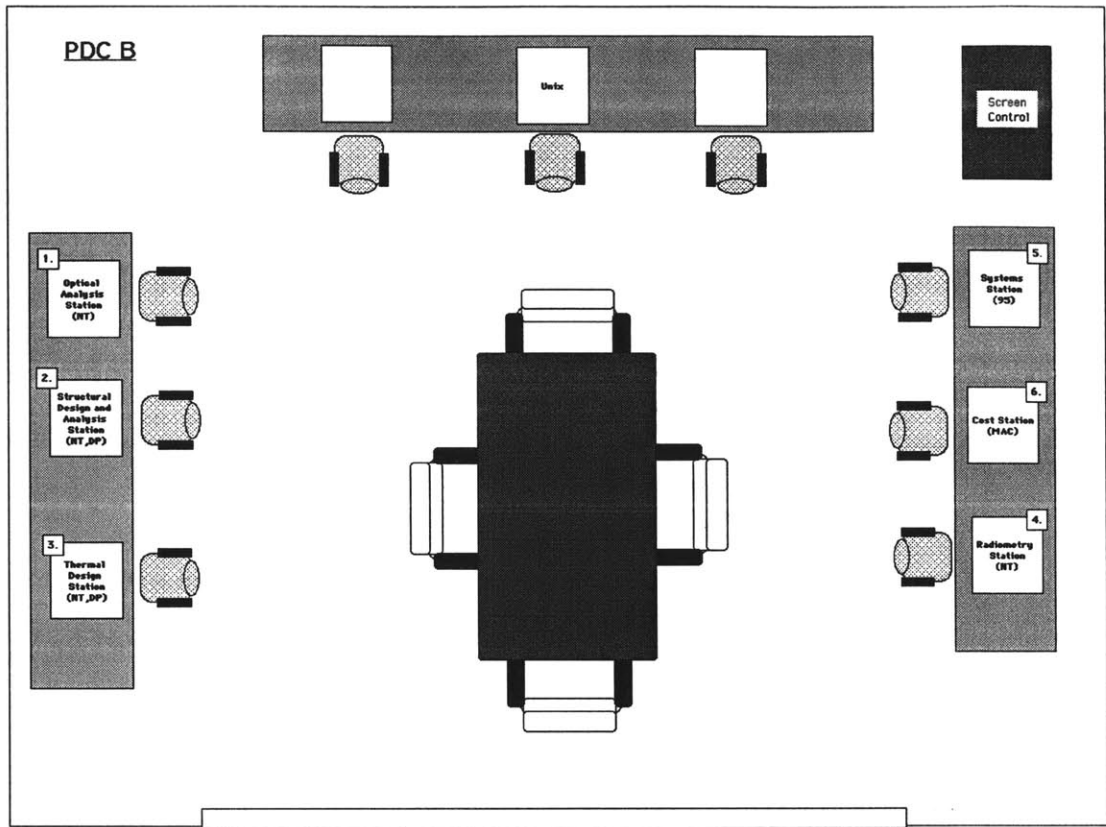


Figure 2.11: The Team-I room architecture (see PDC web site [15])

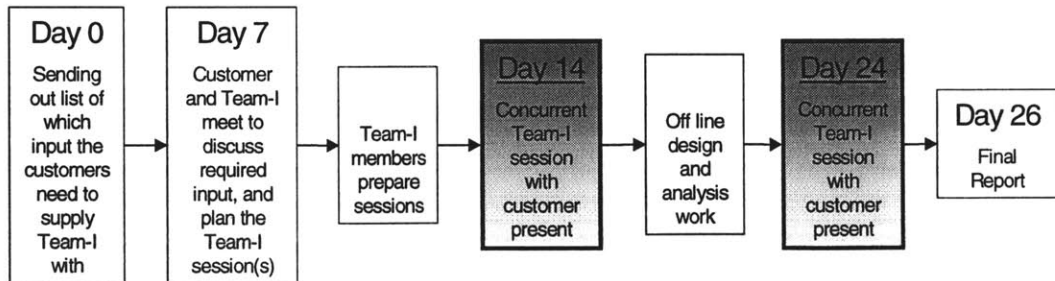


Figure 2.12: Team-I design process scheduling (see Smith [37])

The cost of the development of the capabilities necessary for real-time design can be quite substantial. Developing Team-X capabilities (computers, software and other facility needs) involved over \$1.5 million. Maintenance, operations and upgrades account for about \$500K/year.

The metrics used to evaluate the performance of Team-X are (see Smith [14]):

- Average cost to prepare conceptual mission/spacecraft designs
- Average time to prepare conceptual mission/spacecraft designs
- Number of missions studied per year

The implementation of the PDC, Team-X and Team-I resulted in large improvements in cost, schedule and team performance during the design process. The principal lesson learned was the importance of understanding and improving the design process in order to enhance the productivity of the design teams. Introduction of enhanced software tools or models were not sufficient to produce the results. Managers play a key role in keeping everyone focused and on track. Sometimes, too much automation detaches engineers from how the subsystems are interacting. In the industry context, several months of effort are often required before any significant benefits can be proven using concurrent engineering techniques.

2.3.4 Boeing (Long Beach)

The fourth site visited was Boeing in Long Beach, California. The Multi-Disciplinary Optimization process (MDO) at Boeing (see Figure 2.13) was applied to the design of the new Blended-Wing-Body (BWB) aircraft concept. This process is different from the ones used in the three previous companies because it allows a deeper analysis (preliminary design) of the trade space by optimizing a set of variables using complex optimization algorithms. Instead of satisfying specific requirements with a distinct airframe part, an

array of requirements are satisfied with an integrated airframe (see Wakayama [16]). This process allows for reduced time in the preliminary design from eighteen to three months by automating the study of design trades at a lower level of detail than the conceptual design (the concept has already been defined). Designers can handle more degrees-of-freedom than conventional trade studies at an earlier stage of the design process. It gives a quantitative assessment of the design objectives and constraints.

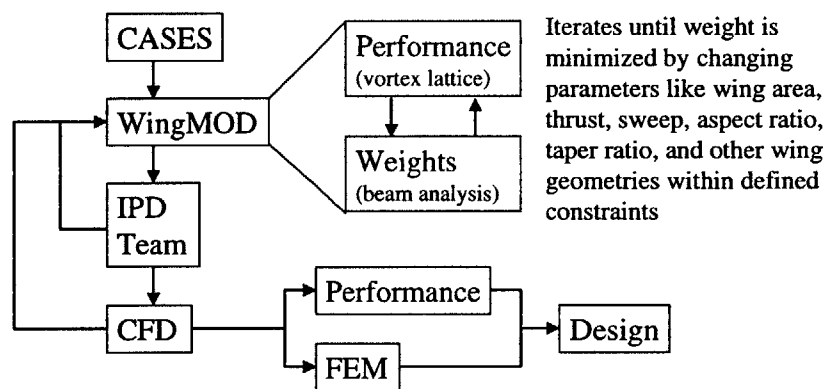


Figure 2.13: MDO process used at Boeing (Long Beach) for wing design.

2.3.5 SAAB

The fifth company visited was SAAB in Linköping, Sweden. SAAB has implemented a company-wide integrated concurrent engineering process covering the whole life cycle of a product to reduce cost, time and number of defects in their aircraft industry. Design tools are distributed and linked throughout the organization. Design centers were created to allow meetings where Integrated Product Development (IPD) teams work together (see Figure 2.14). The objectives of the design centers were to (see Farnworth [17]):

- Decrease the time from design to decision
- Increase the rate of communication for the IPD team members
- Move decision making down to the lowest level possible

- Postpone generating drawings as long as possible
- Capture the design knowledge and carry it forward with the design

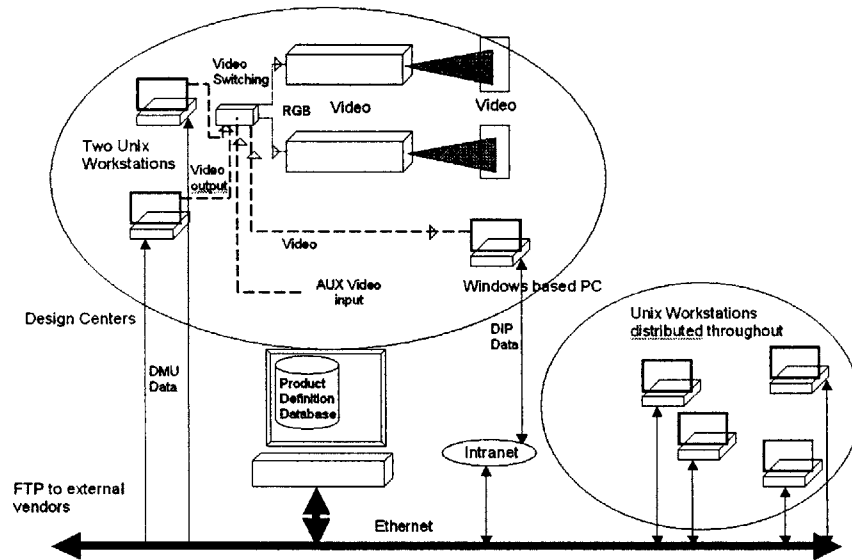


Figure 2.14: The SAAB information system architecture (see Farnworth [17])

Model of data and analysis flow

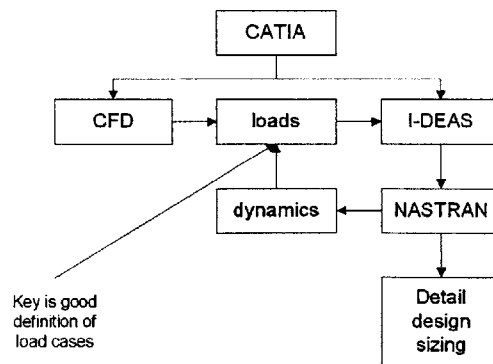


Figure 2.15: SAAB structural analysis and modeling process (see Farnworth [17])

A structural modeling process used to verify sub-assemblies is illustrated in Figure 2.15. A detailed description of the process and design experience is given in Farnworth [17].

2.3.6 Caltech Laboratory for Spacecraft and Mission Design (LSMD)

The last site visited was Caltech's LSMD. Caltech founded the LSMD to provide students with world class educational opportunities in the emerging field of information system assisted design and analysis of complex aerospace systems (see the LSMD web site [18]). The LSMD is used to teach ICE and software development to graduate and undergraduate students. Like the design centers of JPL, Aerospace Corporation and TRW, the facility is equipped with computers around the outside of the room, a large whiteboard at one end, and a large conference table in the middle (see Figure 2.16). The computers are all networked together via a high speed fiber-optic network, and data is stored on the hard drive of one of the computers which has been set up as a server. The facility is at the same time a working and a meeting environment. It is possible to display the content of any monitor on a large screen at one end of the room. Additional capabilities of the facility are two overhead projectors capable of displaying SVGA and VGA, video conferencing (Polycom Viewstation), DVD, and telephonic conferencing. All three different computing platforms (PC, Unix and Macintosh) are available.

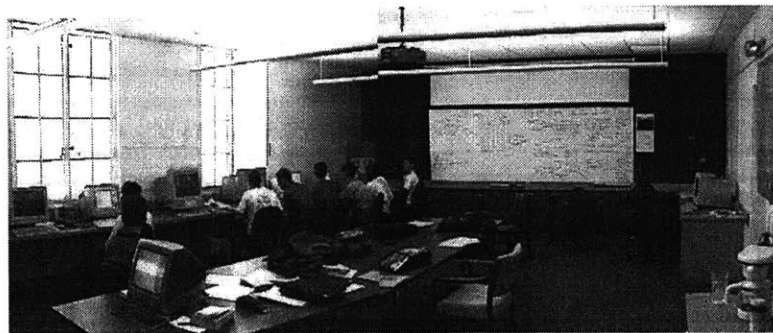


Figure 2.16: Caltech's Laboratory for Spacecraft and Mission Design (LSMD)

The most important course associated with the LSMD is Ae125 Spacecraft and Mission Design. This one year class (three terms) is offered to upper division undergraduate and graduate students. In the class, students learn about the disciplines associated with aerospace and about how information systems can be used in a multidisciplinary environment to enhance team productivity. In the first term, students are exposed to all the major disciplines associated with space flight system and mission design and analysis. The second term covers the principles of system engineering and software development for collaborative design and analysis. Finally, the third term is a collaborative study in which a team of students from varied backgrounds work together to develop a complete end-to-end mission and system architecture, and a conceptual design of a complex system. The architecture typically includes a business case, a technical design and analysis, and an implementation plan. During the fifteen weeks of the last trimester, twelve are used for coding design software module and three for the ICE design study, as opposed to the fifteen weeks of design study required prior to the implementation of the LSMD. A total of fifteen sessions of two to three hours each is generally required to complete the design.

The students flow down a design process to be used during the design sessions each time the course is given. This process is customized to the specific needs of the project, but is representative of the ones used in industry for conceptual design. Each workstation in the LSMD represents a different subsystem. Data exchange between subsystems is achieved using a spreadsheet software named ICEMaker. ICEMaker was developed under an Excel environment. It consists of a “publish and subscribe” tool that allows designers to share and update data in models. ICEMaker is linked to a software named DrawCraft. DrawCraft is a parametric CAD software under development by Caltech that works in a SolidWorks environment. It allows for a rapid visualization of a conceptual design of a spacecraft. It offers templates for common spacecraft components used in conceptual

design. The geometry is driven by parameters entered in ICEMaker. The STEP file format is used to interface with CATIA when more detailed drawings are required. The generation of the requirements is implemented under Unified Modeling Language (UML). UML is an international standard for software development. Other software tools like Pro/Engineer (ProE), SOAP (for orbital analysis), and Matlab are used as well.

The cost to construct the LSMD was \$65,000 with an additional \$20,000 for hardware, and \$100,000 for software development. The maintenance cost is \$1,000 per month.

Physically, the LSMD, which is representative of a design center used in academia, has a very similar architecture to JPL's PDC, TRW's ICDF and Aerospace Corporation's CDC. The software tools used to communicate data are similar as well (linked spreadsheets). However, since the students are writing their own software each time they take the class Ae 125 Spacecraft and Mission Design, the LSMD provides more software development tools and support (like UML). In industry, the software development phase does not occur inside the laboratory. The software modules used in the design centers in industry are customized for a certain type of project (mission design, instrument design, etc.), while the ones developed by students are customized for a particular project since they will be used only for that project.

2.4 Study of Design Processes Covering the Entire Life Cycle

This section is intended to present design processes covering the entire life cycle of a product that could be used in DE-ICE for class purposes. Two design processes were considered: NASA's program/project life cycle and Ulrich and Eppinger's product development process. Comparisons are made with the process followed during the Spring 2000 term in the class 16.89 Space Systems Engineering.

For the management of major systems, NASA uses the program/project life cycle illustrated in Figure 2.17, which consists of a categorization of everything that should be done to accomplish a project. Each distinct phase of a project is separated by control gates.

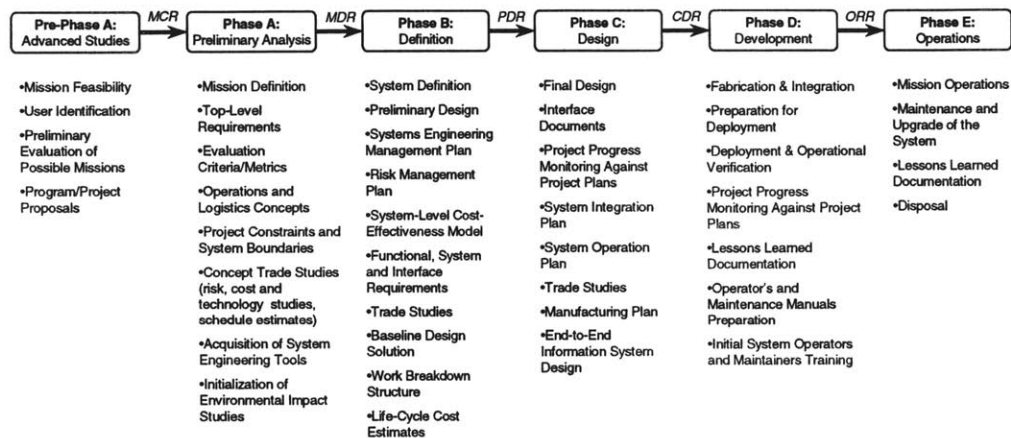


Figure 2.17: NASA program/project life cycle (see NASA Systems Engineering Handbook [19])

Following is the purpose of each phase:

- Pre-Phase A, Advanced Studies: To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected.
- Phase A, Preliminary Analysis: To determine the feasibility and desirability of a suggested new major system and its compatibility with NASA's strategic plans.
- Phase B, Definition: To define the project in enough detail to establish an initial baseline capable of meeting mission needs.
- Phase C, Design: To complete the detailed design of the system (and its associated subsystems, including its operations systems).
- Phase D, Development: To build the subsystems (including the operations system) and integrate them to create the system, meanwhile developing confidence that it

will be able to meet the system requirements, then to deploy the system and ensure that it is ready for operations.

- Phase E, Operations: To actually meet the initially identified need or to grasp the opportunity, then to dispose of the system in a responsible manner.

However, even if each phase is distinct, it is possible that the same activity occurs in more than one phase. For example, different levels of trade studies are occurring during Phase A (Preliminary Analysis), Phase B (Definition) and Phase C (Design). Project Monitoring Against Project Plans is occurring in Phase C (Design) and in Phase D (Development). The documentation produced in one phase is also verified and refined in the following phase. A complete set of activities occurring in each phase is listed in (see NASA Systems Engineering Handbook [19]).

Ulrich and Eppinger [20] suggested the product development process illustrated in Figure 2.18. This process is more adapted to a specific product than to a program or a project. It covers most of the life cycle of a product, from the mission statement to the launch of a product.

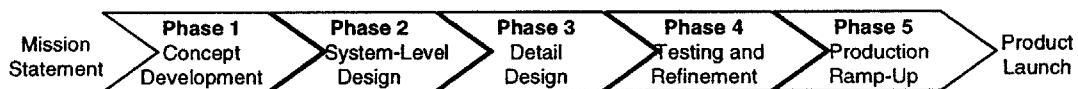


Figure 2.18: Ulrich and Eppinger’s product development process (see Ulrich [20])

Activities performed in each phase are mentioned here:

- Concept Development: The concept development phase requires a lot of integration among the functions performed by the development team. This phase has been expanded and the set of activities achieved are illustrated in Figure 2.19.

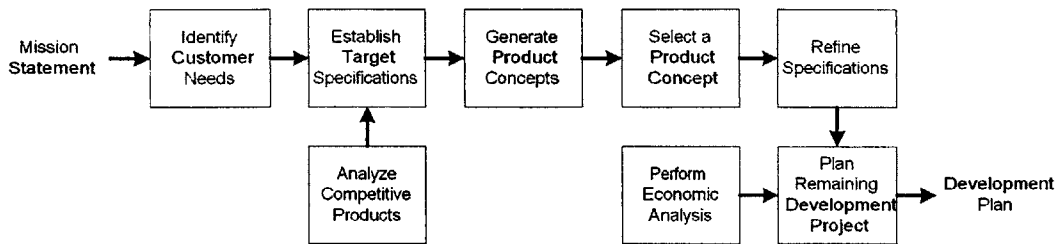


Figure 2.19: Activities performed during the concept development phase (see Ulrich [20])

- **System-Level Design:** This phase includes the definition of the product architecture, the division of the product into subsystems and components, and generally the final assembly scheme for the production system. This phase output a geometric layout of the product, functional specifications for each of the product subsystems, and a preliminary process flow diagram for the final assembly process.
- **Detail Design:** This phase includes the complete specification of the geometry, materials, and tolerances of all the parts. The production process is established and tooling is designed for each part. The output of this phase is the complete geometric description of each part and its production tooling, the specifications of the purchased parts, and the process plans for the fabrication and assembly of the product.
- **Testing and Refinement:** This phase involves the construction and evaluation of multiple preproduction versions of the product. Prototypes are first tested to determine whether or not the product will work as designed and will satisfy the customer needs. Then, the prototypes are tested by customers in their own use environment to verify performance and reliability in order to identify changes for the final product.
- **Production Ramp-Up:** This phase involves the production of the product using the intended production system. The work force is trained and any remaining problems

in the production process are solved. The transition from production ramp-up to on-going production is gradual and continuous, and the product is launched sometime during this transition.

The process used by the students for the project of the class 16.89 Space Systems Engineering is illustrated in Figure 2.20. This process was mapped during and after the class. The students started the design having a minimum of requirements. The project was divided into three main phases: the mission definition & preliminary analysis, the preliminary design and the detailed design. Activities were conducted concurrently in each of the phases. Limited interactions between the groups occurred in the preliminary design phase and difficulties were encountered. ICE sessions were conducted at the beginning of the detailed design phase instead of the preliminary design phase after noticing the communication problem between the groups.

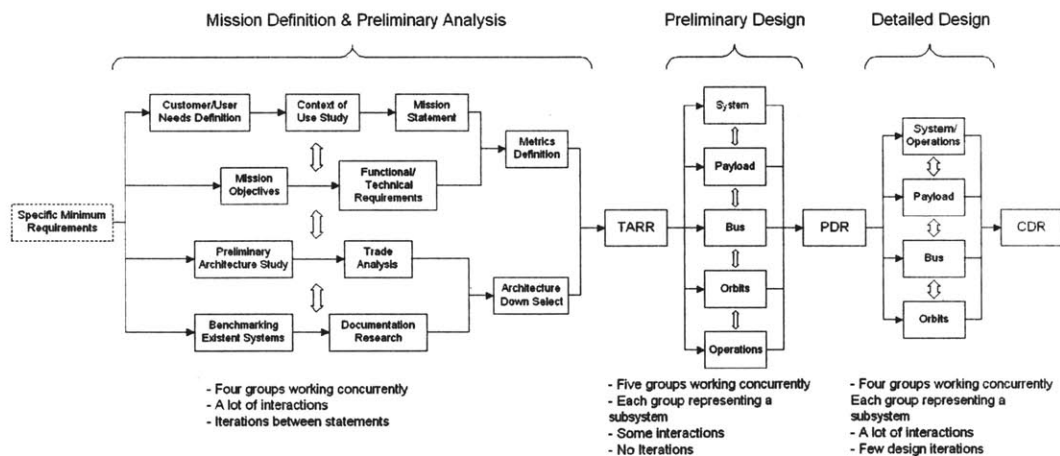


Figure 2.20: Process used in 16.89 Space Systems Engineering during Spring 2000

All three design processes described present similarities. NASA's program/project life cycle covers the entire life cycle of a project and could be used as a template. Figure 2.21

summarizes, compares and matches each of the different phases of the design processes previously mentioned.

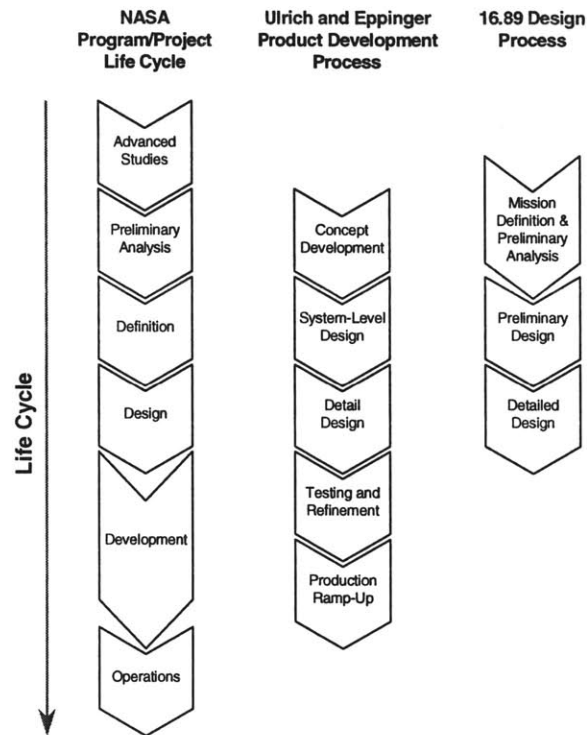


Figure 2.21: Comparison of the design processes

2.5 Chapter Summary

This chapter presented an overview of existing research and experience acquired by academia, government and industry both in integrated concurrent engineering and in distance collaboration. Experiences and research achieved in the following four subsections of interest were presented: 1) teaching engineering design and active learning, 2) facilities and technologies for teaching engineering design, 3) teaching distance courses and distance collaboration, and 4) integrated concurrent engineering in industry and government.

Numerous sites practicing integrated concurrent engineering were visited (TRW, Aerospace Corporation, JPL, Boeing, SAAB and Caltech). The ICE process used by most

of these sites is summarized in Figure 2.22 and Figure 2.23. Design sessions usually take from two to three hours, and occur no more than three times per week in industry (Caltech practices one session per day during three weeks to complete a design).

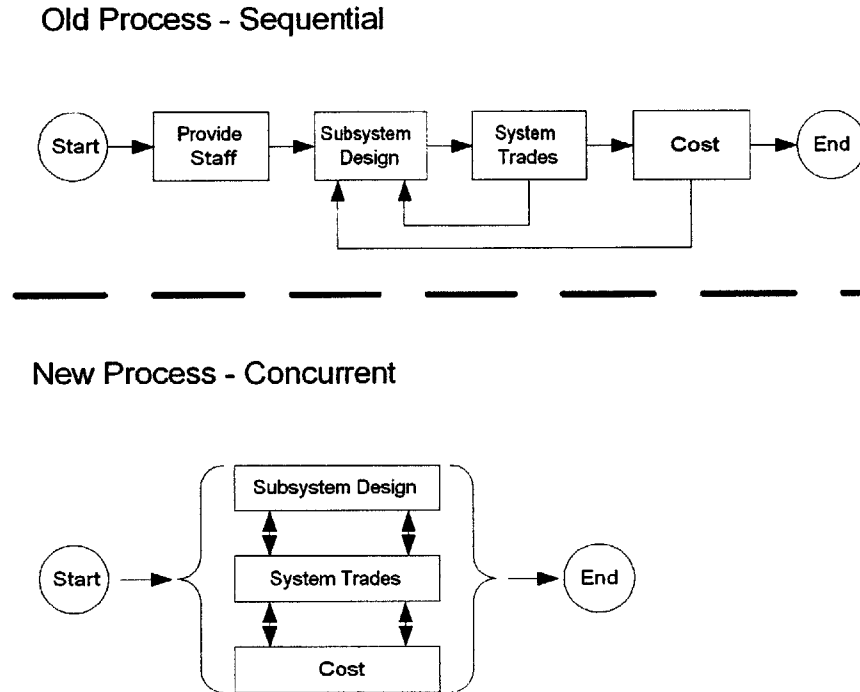


Figure 2.22: Comparison between old and new conceptual design processes (see Oberto [38])

Lessons have been learned while developing and operating these design centers. The understanding of the design process is very important in order to practice ICE. Following a detailed script during the design sessions helps to keep all participants on track. Optimization techniques are generally not used for conceptual design. During conceptual design sessions, designers should not search for optimal designs, but for a design that meets all mission requirements at acceptable costs and risks. Face-to-face communication is very

important, and faster than E-mail. Preparations, like research and tool development, need to be done prior to the sessions.

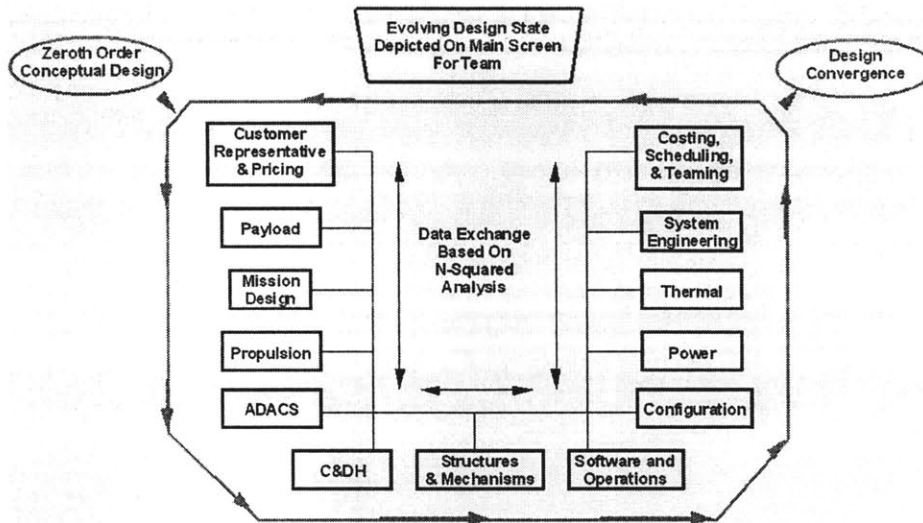


Figure 2.23: ICE process used in LSMD for conceptual design (see Sercel [39])

Finally, design processes covering the whole life cycle of a project/program/product were introduced and compared with the process followed in the class 16.89 Space Systems Engineering. NASA program/project life cycle is the one covering the most of the life cycle of a project and therefore is the one recommended by the author to be used as a template for design classes.

Chapter 3

Education Needs Study For DE-ICE

3.1 Chapter Overview

The purpose of this chapter is to investigate the MIT A/A department's needs for an educational design environment for integrated concurrent engineering (ICE). The academic environment is characterized first. Then, the results from research performed during the Spring 2000 semester with faculty members teaching pilot design courses is presented, as well as the results of a survey distributed throughout the A/A department. The needs for DE-ICE as defined by the team, and the constraints that the team had to deal with are introduced. Finally, the experience acquired from two design classes during the Spring 2000 semester is used to justify the needs for DE-ICE.

3.2 Characterization of the Academic Environment

Academia involves different types of people, achieves different goals and has different time constraints than industry. It is important to characterize this academic environment before defining the needs for DE-ICE.

The main difference between industry and academia is obviously their end goal. The ultimate goal of industry could be perceived as making a profit, while the ultimate goal of academia is to teach students. As seen previously, industry's goal in implementing design centers is to lower cost and time required to design a product or a mission, and to increase the quality of the design. Academia is implementing design centers to teach students the product design/development process and to facilitate learning in the research environment.

Another difference between engineers in industry and students in universities is the level of experience. Engineers in industry, especially the ones that are working in design centers, have years to decades of engineering experience. They know how to efficiently

approach a problem, and they often have developed a network of contacts with other engineers for collaborative purposes. In comparison, students spend a lot of time learning engineering fundamentals, theory and problem solving, and require more technical support than engineers in industry.

The time allocated to develop tools in a class is a lot different than in industry. Engineers in industry can afford to take the time to develop many customized tools since they use them for months, probably years. Students spend less time to develop customized tools since they use them for one or two semesters for a particular project. Therefore, the tools developed are less efficient and less powerful than the ones in industry. They are less user-friendly, and often require the developer to be present in order to use them.

The variety of projects performed in academia is much broader than in a specific industry. To illustrate this thought, one could consider the projects achieved by MIT A/A department and the ones achieved by JPL's Team-X and Team-I. Students at MIT A/A department are performing aeronautics related projects (Lighter-Than-Air vehicle design project in 16.00 Introduction to Aerospace Engineering and Design, for example) as well as aerospace related projects (design of a satellite constellation around Mars to provide navigation and communication to Mars ground assets in 16.89 Space Systems Engineering, for example). Team-X at JPL is performing only space missions/spacecraft design while Team-I focuses only space instrument design (see section 2.3.3). Design environments in academia have to account for the project diversity to be performed, and the experience level of the learner.

3.3 Feedback on Design Pilot Courses

Interviews with faculty members teaching design classes were performed from March through April 2000 to investigate how design classes are taught in the A/A department.

Professors Dava Newman, David Miller, Mark Drela, Daniel Hastings, Peter Young, John Hansman and Kim Blair were interviewed. The questions asked during the interviews and the answers provided are given below:

1. What are you teaching in the class?

Faculty interviewed were developing and teaching both undergraduate and graduate design courses. The design project subjects varied significantly (Lighter-Than-Air vehicle, heavy mass airplane, Precipitation Measuring Instrument for a Space System (PreMISS), Synchronized Position Hold Engage Re-orient Experimental Satellites (SPHERES), satellite constellation around Mars to provide navigation and communication to Mars ground assets (Minerva), and projects involving business issues). In some classes, the design projects differ from one year to another. For example, in 16.89, the previous year's (Spring 1999) project involved the design of a system to search for terrestrial planets and related origins (the ASTRO project). In Spring 2000, the Minerva design project for 16.89 involved the design of a satellite constellation around Mars to provide communication and navigation to Mars ground assets. Design projects last from one to three semesters. The longer time frame allows coverage of the entire life cycle of a product and involves fabricating engineering hardware.

2. How are you planning to use the new laboratory (teaching, distance teaching, design laboratory, distributed classroom, etc.)?

All the faculty members interviewed are planning to lecture in DE-ICE and the Learning Laboratory facilities. Some are even planning to do distance lecturing or to attend lectures through the video conferencing system. Most of the instructors wanted to use DE-ICE as a laboratory for their students to perform design projects and other team related work. Some faculty required DE-ICE to have the capability

to switch rapidly from a lecturing environment to a laboratory type environment for team work. Some would like to have the technical ability to perform simulations, demonstrations, and to interact easily with external facilities like machine shops or laboratories.

3. How do you plan to teach to the class (special focus on activities to enhance active learning)?

The first response to this question was obviously that performing a design project is a form of active learning. Requiring the students 1) to interact with other students and faculty during the lectures and the design phases, and 2) to present the final design, was seen as a very effective way to achieve active learning. Some faculty want their students to develop design portfolios or a notebook describing their experience, the design process and their contributions to the project. The use of multimedia and interactive CD-ROMs was envisioned to achieve demonstrations, real-time simulations and web-based exercises, as well as for the students to easily access teaching material. The use of Lego building blocks was even suggested.

4. What is the design process you intend to follow?

Most of the faculty interviewed prefer to require specific project milestones to be achieved instead of a specific design process to be followed to give more freedom to the students. They set up specific design reviews throughout the semester, and the students manage to achieve the required reviews on time. Faculty are not rigid and vary in the way they teach the process of design. Graduate students get high level statements at the beginning of the semester, and then flow down their own design process. Undergraduate students, since they may not be experienced enough to flow down their own process, receive more inputs in earlier steps of the

design. Quality Function Deployment (QFD) might be taught, as well as other suggested design processes (see Figure 3.1).

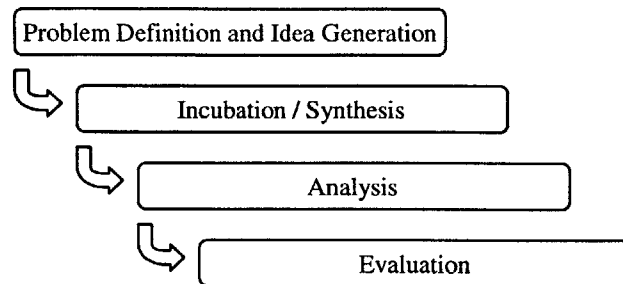


Figure 3.1: Design process suggested in 16.00

5. What type and kind of guidance are you planning to give your students?

Most of the faculty give more lectures at the beginning of the semester to present theory needed later in the semester, and then let the students do more work in class. Some will give on-demand lectures after that, depending on how the project evolves, trying to be as flexible as possible. A lot of feedback is always given to the students after their design review presentations. However, more guidance is provided to undergraduate students than graduate students. Faculty plan their work a long time ahead with respect to the schedule. Class notes are distributed regularly, and sometimes previous designs done by students in past years are provided. In some classes, each subsystem group has a mentor or a staff member to work with them on their chosen speciality. More flexibility is given to graduate students and in some cases faculty just let them design free form.

6. What software tools do you ask your students to use?

Similar software to what was used in the A/A department's design laboratory during Spring 2000 is required:

- Access to the Internet, Athena and the A/A department's design server

- Microsoft Office tools (Word, Excel, PowerPoint, Access, FrontPage)
- Web-based tools (Internet Explorer, Netscape, Dreamweaver)
- CAD tools (ProE, Working Model, Adobe Photoshop)
- Programming language (Matlab, Maple, FORTRAN compiler, AVL code (Professor Drela's code), NASTRAN)
- Video conferencing tools (NetMeeting)
- Simulation and demonstration tools (SOAP, Macromedia Director)
- Specialized tools (STK for orbital analysis, Orcad for electrical circuit layout, ICE-Maker for design session management)

7. Any other suggestions?

Having phones and lockers available for students in DE-ICE would help. One faculty member suggested having a place to store printouts. Another faculty member is more interested in the plan on how to use DE-ICE than in the tools and hardware available. He insisted that he will need somebody responsible for collaborative design, simulations, etc. This person could be a teaching assistant, for example. It was suggested to have CD-ROM lectures available as backup resources for students. Another faculty member mentioned that it would be good to find a way to reduce noise when working in teams, maybe through movable walls.

3.4 Data Analysis of the DE-ICE Users Experience Survey

A web survey was performed at the beginning of the semester. The goal of the survey was to better understand the experience of the potential users of DE-ICE in design and working with design tools, as well as their communication practices. Over 50 faculty/staff members, graduate and undergraduate students completed the survey. A copy of the survey along with detailed results are provided in Appendix C.

The results show that the most popular operating system for the students is PC-based (NT), while the faculty prefer the Macintosh platform. The most common software tool is MS Office (Word, Excel, PowerPoint, Access, FrontPage) for both faculty and students. Matlab is the most common mathematical tool, being very important for the students. ProE is the most popular CAD/CAM tool, followed by Autocad and I-DEAS. Microsoft Visio 2000 was found to be quite important for graduate students. FORTRAN and Java were the most used programming languages.

Software and system usability was found to be the most important factor influencing the efficiency of the design process for faculty, and one of the most important factors for the students. Communication with team members is critical for the students, and the ability to share data easily is very important for graduate students. The preferred communication method is still face-to-face communication for everybody, followed by E-mail. But the preferred methods for communicating ideas, concepts and factual information are meetings, E-mail, sketches and presentations. The difficulty of representing ideas in an E-mail, and the varying interpretation of sketches (when the designer is not present to explain the sketch) were found to be the most limiting factors of electronic communications.

The most important factors limiting the ability to share engineering data/information are the different file formats used by the design tools and the speed of the network. Undergraduate students consider the location of the team members to be an important factor while graduate students put more importance on the system interfaces.

Understanding the models used in the environment and their limits makes faculty more inclined to use an integrated engineering environment. The undergraduate students rank the time saved by reducing repetitive tasks important, as well as the overall time

needed to develop a design. The graduate students would like to have a system enabling data integration, data access, and ease of use.

3.5 Needs Definition and Classification

One of the most difficult challenges that the DE-ICE team faced was to identify, define, rationalize, group and order the needs of the A/A department concerning DE-ICE while collecting data from the different end users (see Chapter 2 and the beginning of Chapter 3).

To start the iteration process, three main user needs were identified:

1. The system should be scalable for varying levels of user experience and training
2. The system should enable a distributed design team to concurrently design aerospace systems
3. The system should add value to the engineering project and enhance previously learned knowledge

After iterations and consultations with faculty members and staff, the three user needs were expanded into nine specific needs, split into two categories: pedagogical needs (boxes 1 to 4 in Figure 3.2) and operational needs (boxes 5 to 9 in Figure 3.2).

Each of these needs has been defined:

1. Active learning in a design environment

The design of a complex system cannot be easily taught solely by passive learning. Other learning methods in which the students are encouraged to participate in their own learning are used (see section 2.2.1). Learning modes that can be used are:

- ◆Content experiential (i.e., hands-on experimental)
- ◆Project-Based Learning
- ◆Case-Based Learning

- ◆Chalk talk with class discussion, think-pair-share
- ◆Self-Directed (interactive CD-ROM, Internet)

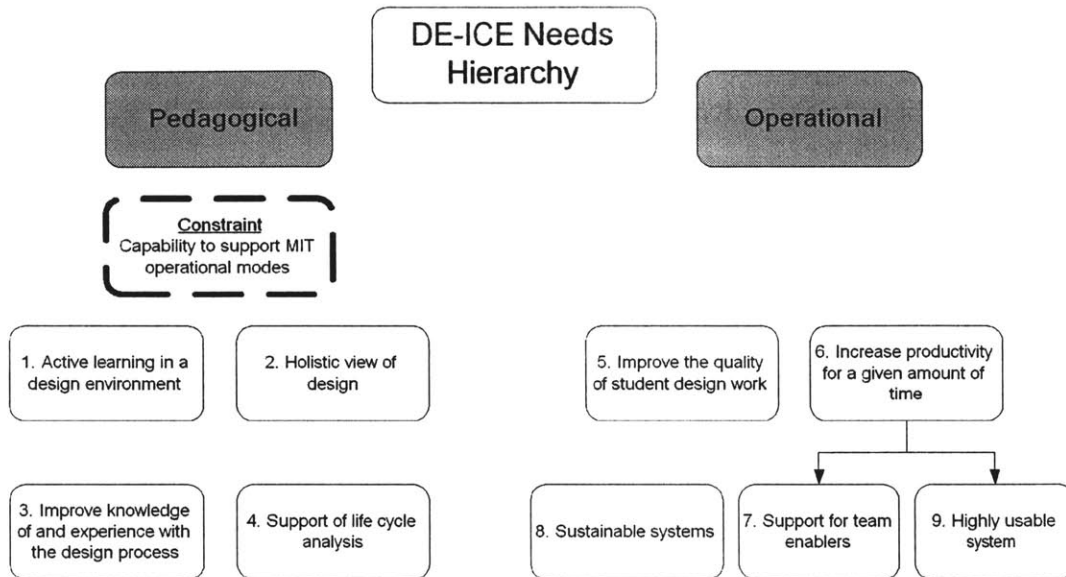


Figure 3.2: DE-ICE user needs

2. Holistic view of design

Design entails understanding the needs of the customer, conception, analysis, iteration, planning, coordination, and communication. Designing a system while taking into account and understanding the life cycle costs and the risk issues of a system are key to the design of modern systems. DE-ICE should enable students to:

- ◆Perform system-wide design analysis and trade studies
- ◆Better understand the effect of a modification of a subsystem on the overall system
- ◆Better understand the influences of the design decisions on manufacturing, assembly and operation of a system

3. Improved knowledge of and experience with the design process

By utilizing the DE-ICE system, the hope is that students should gain knowledge and be enabled to make mature decisions early in a design process. Simulation and modeling should be provided to allow students to establish estimates of the behavior of a system (i.e. dynamics, thermal, structural, etc.), costs, performance, manufacturability, assembly, and operations.

4. Support of life cycle analysis

The system should be able to support any phase of the design: needs, requirements, functional and behavioral analysis, interface specification, functional-to-physical mapping, testing and verification planning, detailed design, digital mock-ups, manufacturing and assembly, variational simulation analysis, operational planning and support, data collection, and disposal.

5. Improved quality of student design work

The system should improve the quality (i.e.: depth of analysis, manufacturability) of designs by allowing users to consider the downstream influences on the design (fulfilling the requirements, depth of analysis, reducing reworks, ease of manufacture, delighting the customer, etc.). The system should provide a method to analyze the product and the design/manufacturing process and evaluate their fit. By integrating models and simulations, DE-ICE should enable students to iterate a design to achieve a much higher quality design than what could be done previously when team members were working on separate models.

6. Increased productivity for given amount of time

The system should increase the productivity of the teams so that more time can be spent on the design and not wasted learning new software or performing repetitive

tasks. Users should be able to make changes to any part of the design and see the effects automatically. Drawings, charts, graphs, and diagrams should be easily extracted for use in reports, presentations, or electronic discussions.

7. Support for team enablers

Since the teams may be non-located, gathering the design team members for meetings or reviews may be impractical. Therefore the system should support electronic meeting services where information can easily be shared. The system should also have the capability for audio and video communication so non-located team members, guest lecturers or expert help can communicate.

8. Sustainable system

Features of the system should be easily configured to work in a stand-alone component design or configured to any users' needs. The system should be easily maintained and upgraded through the resources available within the MIT community. The systems capability should evolve with the users' needs and with time. Users should be able and encouraged to make changes to the system for its improvement.

9. Highly usable system

The system should be intuitive so as not to add complexity to the design process. Users should be able to operate the system on any computer platform. They should be able to use familiar tools and software interfaces when working on the system. Integrating data into a model or running a global optimization on the system should be simple tasks and not be time- or effort-intensive. Students should be able to work on the design from any location and not be constrained to the lab.

3.6 Constraints

The MIT A/A department foresees itself operating in many types of educational modes. Approximately twenty operational modes ranging from teaching in labs to paper designs were defined. The complete list and definitions of the modes can be found in Appendix D. The DE-ICE system should be designed to support a number of these modes, principally:

- **Design project mode**

Class-based design projects that are carried out over the period of a semester by student teams from a given class

- **Lecture/presentation mode**

Standard class mode, where a lecturer discusses/presents class material (boards, web, overheads, etc.)

Also relevant to these modes are:

- **Large systems mode**

Team-oriented multidisciplinary programs that are very design-intensive and that could include systems prototyping

- **Collaborative project mode**

The A/A department working with other universities, government and industry

- **Research design support mode**

Used by research teams to work through a segment of the research design (analysis, design, communications, presentations)

- **Distance learning/teaching mode**

Video-conferencing, broadcasting classes/presentations, receiving classes/presentations

- **Interactive electronic class mode**

Fully electronic classroom where students are able to do computer-based work in real-time, and to present a certain project

Supporting these modes was the main constraint that the team had to respect. It was defined as a pedagogical constraint (see the constraint box in Figure 3.2).

3.7 Education Case Study

Two design classes were conducted in the A/A department during Spring 2000: 16.89 Space Systems Engineering and 16.982 Aircraft Systems Engineering. Space Systems Engineering involved the design of a constellation of satellites around Mars while Aircraft Systems Engineering focused more on business issues surrounding the design of a very large aircraft. Both of these classes faced some difficulties during the semester.

Space Systems Engineering (16.89) was a technical design oriented class. The class was designed so that interactions would occur with students taking a similar class in California Institute of Technology. In fact, as described in Nolet [31], the level of interactions attained was weak. One of the reasons for that was the lack of a facility in the A/A department where students could do distance collaborative design with students from another university. The laboratory temporary used during the construction of the Learning Laboratory was very small and equipped with very limited video conferencing capabilities. It wasn't seen as a good environment for distance collaborative design. Students mentioned that they would need a larger and better equipped laboratory. Since this laboratory was the only public PC laboratory for the A/A department, it was often packed with undergraduate students doing their homework.

A *lessons learned* session at the end of Space Systems Engineering reveals valuable information from the students concerning the pedagogical aspects of this design class. Every student agreed that this class added a lot to their experience. They found it painful,

requiring a lot of work, but at the end very valuable. The main problem encountered during the semester was the organizational structure of the class. Faculty did not want to impose too much structure because the course objectives included learning how to work in teams. Students would like to have classes on the design process and teamwork at the beginning of the semester. Another approach to help students would be to give them access, through DE-ICE, to on-demand help, like on-line help, as well as access to previous design work.

Students found the design sessions very valuable to explore the trade space through iterations. But, as mentioned in 2.3.1 - 2.3.3, it was found that a certain level of preparation had to be reached before those design sessions. ICEMaker, the software developed in an Excel environment by Professor Joel Sercel from Caltech, was found to be very helpful, even though a package that connects Matlab and Excel was still missing. More information can be found in the Project Minerva design report [49].

Space Systems Engineering lectures occurred in a room equipped with a PictureTel video conferencing system. The video conferencing system was used when a distance speaker was lecturing to the class. This occurred four times during the semester. The video conferencing system allowed access to engineers in industry who would not have been able to travel to MIT to deliver their lectures. Access to engineers in industry was found very valuable.

Aircraft Systems Engineering (16.982) was less technical than 16.89 and was focused more on aircraft business design issues. All the lectures were given to local and distant students simultaneously. The lectures occurred in a room equipped with a PictureTel video conferencing system. Some technical problems concerning reliability, time delays, audio and visual quality, and synchronization were identified in Nolet [31]. Once again, students were very pleased to have experienced engineers on their team. That would not have been

possible without the video conferencing system. On the other hand, students identified some improvements, other than the technical issues previously mentioned, that should be implemented in the future DE-ICE. DE-ICE should not require a professional technician to operate the facility as is the case with the current facilities at MIT in building 9 (rooms 9-151, 9-152 and 9-057). Faculty members and teaching assistants ideally should be able to operate the system by themselves. The system used in building 9 is quite expensive (more than \$250 per hour) and is not available for students to work with their distant colleagues.

3.8 Chapter Summary

The purpose of this chapter was to investigate the MIT A/A department's needs for an educational design environment for integrated concurrent engineering (ICE). The academic environment was characterized and differences between industry and academia were highlighted. Specifically, differences occur in the:

- End goal
- Level of experience
- Time allocated to develop tools
- Variety of projects performed

The results from research performed during the Spring 2000 semester with faculty members teaching pilot design courses were presented. The topics of interest were:

- Design course content
- Methods of using the laboratory
- Methods of teaching the class
- Design process followed in MIT design courses
- Guidance given to students

- Software tools used in design classes

The results from a survey to better understand the experience of the potential users of DE-ICE in design and working with design tools, as well as their communication practices, are summarized in Table 3.1.

Table 3.1: Survey data summary

	Undergraduate Students	Graduate Students	Faculty Members
Preferred operating system	PC	PC (NT)	Mac
Most popular software tools	Matlab MS Office ProE Basic	Matlab MS Office Mathematica FORTRAN MS Visio ProE	MS Office Matlab FORTRAN NASTRAN
Factor influencing the effectiveness and efficiency of the design process	Communication with team members Software and system usability	Ability to share data easily Communication with team members	Software and system usability Graphical capability and visualization of data
Preferred communication method	Face to face E-mail	Face to face E-mail	Face to face E-mail
Preferred method to communicate ideas and concepts to team members	E-mail Sketches Meetings	Sketches E-mail Meetings	Meetings E-mail Sketches
Preferred method to communicate factual information to co-workers/team members	E-mail Meetings Sketches	Meetings E-mail Presentations	E-mail Meetings Presentations
Limiting factors of preferred communication method	Difficulty representing ideas in E-mail Difficulty representing ideas in sketches	Varying interpretation of sketches Difficulty representing ideas in E-mail	Difficulty representing ideas in E-mail Varying interpretation of sketches
Factors limiting the ability to share engineering data/information	Network speed Location of the team members	Network speed Different file formats	Different file formats
Factors encouraging use of DE-ICE	Reducing repetitive tasks Reducing design time	Allowing data access and integration Easy to use	Understanding the models Understanding the limits of the models

Nine educational needs and one constraint were identified, defined and classified:

1. Active learning in a design environment
2. Holistic view of design
3. Improved knowledge of and experience with the design process
4. Support of life cycle analysis
5. Improved quality of student design work
6. Increased productivity for given amount of time
7. Support for team enablers
8. Sustainable system
9. Highly usable system

CONSTRAINT: Capability to support MIT operational modes

Finally, the experience acquired in two design courses during the Spring 2000 semester (16.89 Space Systems Engineering and 16.982 Aircraft Systems Engineering) and the lessons learned were presented.

Chapter 4

Design Requirements

4.1 Chapter Overview

This chapter presents the technical requirements for DE-ICE that have been derived from the needs previously discussed in Chapter 3. These requirements have been ranked in order of importance using a Quality Function Deployment (QFD) matrix. Conflicts between the most important requirements are described. A use case study which helped to identify any missing requirements and the tasks that each different type of users can achieve through DE-ICE is presented.

4.2 Technical Requirements Selection

The methodology used to develop and select the main technical requirements for DE-ICE is based on the needs presented in Chapter 3. To minimize human biases, a QFD matrix (see Boppe [50]) was used (see Figure 4.1). First, the needs were sorted by category (pedagogical or operational) in the left column of the QFD matrix. Then, a weighting factor determined by the team and the advisors was applied to each one (10 = high priority, 1 = very low priority). For each need, a set of technical requirements that satisfy it was developed. Then, the list of requirements established was thoroughly discussed and some requirements were grouped together. The final list of requirements retained can be found in the top row of the QFD matrix of Figure 4.1. The next step was to quantify the relation between each of these technical requirements and each of the needs. The following factors were used: 9 = strong relation, 3 = moderate relation, 1 = weak relation, and zero (or null) meant no relation at all. The absolute scores were computed by multiplying the relation factor by the weighting factor of the need assessed, and by adding the scores obtained for each relationship for a particular technical requirement.

The absolute scores for each technical requirement (TR) are displayed in the *TR Importance Weight* row in the QFD matrix, and the relative scores are displayed in the *Relative Weight (%)* row.

4.3 Requirements Prioritization

From the results obtained with the QFD matrix of Figure 4.1, the three technical requirements having the highest score were first selected. A sensitivity analysis (by varying the factor representing the relationship) was performed to see the effect on the absolute scores. The results of the sensitivity analysis convinced the team to retain two other technical requirements. The first five requirements retained are defined as:

1. Design and analysis support

Provide the hardware and software tools needed during the design and analysis process.

2. Provide guidance throughout the design process

Provide a roadmap through the life cycle of a product by giving access to information, examples, theory, etc.

3. Planning and management of the design process

Help to define, schedule and monitor tasks, deliverables between tasks and the progress of the project.

4. Experimentation support

Support design, implementation, analysis and integration of the results of experiments, including resources and tools that help to interface with computers, machines, controllers, etc.

5. Operate on any platform

Provide the ability to work in any environment (Unix, Mac, PC operating systems).

Three more technical requirements were selected because of their strong relation (relation factor of 9, see QFD matrix in Figure 4.1) with the constraint (capability to support MIT operational modes). They are defined as:

6. Distance collaborative support

Provide the capability to remotely attend a class, easily interact with remote teammates and access external resources (specialists, professors, etc.).

7. Flexible system

Have the ability to adapt easily to different operational modes, subjects, projects or new technology.

8. Presentation and reporting support

Provide students the capability to effectively present and report their progress and results, and to give feedback.

Some metrics have been developed to assess each of the requirements after DE-ICE becomes operational (see the *Units* row in the QFD matrix). Difficulties were encountered in finding metrics to evaluate the pedagogical requirements, and improvements to the process are recommended for future work. Comparisons with existing systems have been made by evaluating the main technical requirements from the metrics selected (see the *Benchmarking* rows in the QFD matrix).

4.4 Conflicts

Four main conflicts were identified between the eight technical requirements detailed. The conflicting technical requirements are:

- *Design and analysis support* and *flexible system*

Design and analysis tools are usually complex software requiring specific hardware types in order to work optimally. They are often not compatible with each other, which might cause problems when upgrading the system. The large amount of memory required might limit the simultaneous use of other software.

- *Design and analysis support and operate on any platform*

Few design and analysis tools work on all three major platforms (PC, Mac and Unix). Compatibility issues have to be considered when inputting a result from one platform to another. Software has to be chosen carefully.

- *Distance collaborative support and flexible system*

Most of the distance collaborative tools have limited capability in terms of ease of use, application sharing and mode of communication. This might affect the flexibility of the system. The distance collaborative system to be implemented shouldn't limit the capability of the students to work in teams or to communicate with their distant colleagues. The lecturer should be able to use any mode of communication necessary (speech, slides, sketches, demonstrations, etc.).

- *Distance collaborative support and operate on any platform*

The distance collaborative system to be chosen has to be compatible with all of the three major operating systems so that no constraint is imposed on the users of the system.

A complete QFD matrix with all the conflicts identified can be found in Appendix E.

4.5 Use Case Study

To characterize and illustrate the way that users or other entities (computer, hardware, etc.) directly interact with the system (see Boppe [50]), a use case study was undertaken. Two main diagrams were drawn: the first representing the interaction between the main users

(students, faculty, teaching assistants (TAs), staff and external users) and the system (see Figure 4.2), and the second representing and expansion of a specific function of the first diagram (see Figure 4.3).

On the first diagram, dashed lines represent tasks that flow from other tasks instead of users. Tasks can be achieved by more than one user type. From this diagram, it can be seen that the system itself has to interact with external facilities (wind tunnel and manufacturing laboratory). The diagram helps define software/hardware accessibility once the system is implemented. Faculty and teaching assistants (TAs), even if they are not performing all the tasks, should have the ability to perform all functions needed during a class.

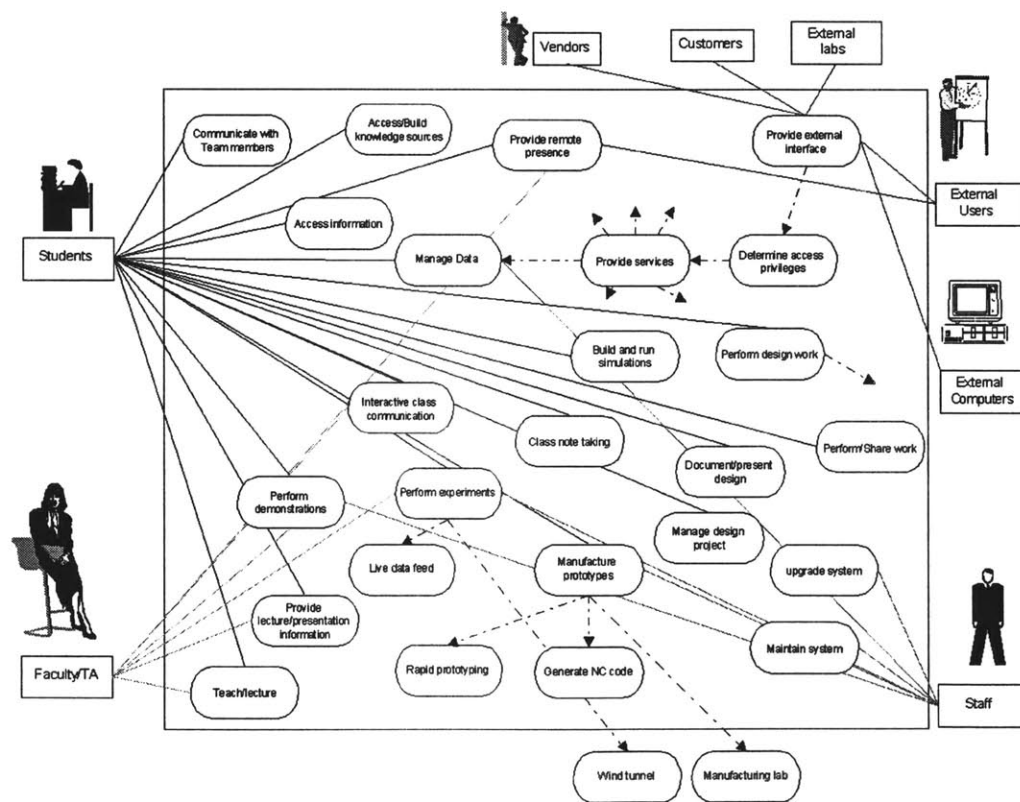


Figure 4.2: DE-ICE use case diagram

The function *Perform design work* in the DE-ICE use case diagram has been expanded into the *Design use case diagram* of Figure 4.3. This diagram gives more details concerning the design tasks achieved by the users. It is possible for the users to have their data managed by a computer through a data managing software like DOME (see 2.2.2).

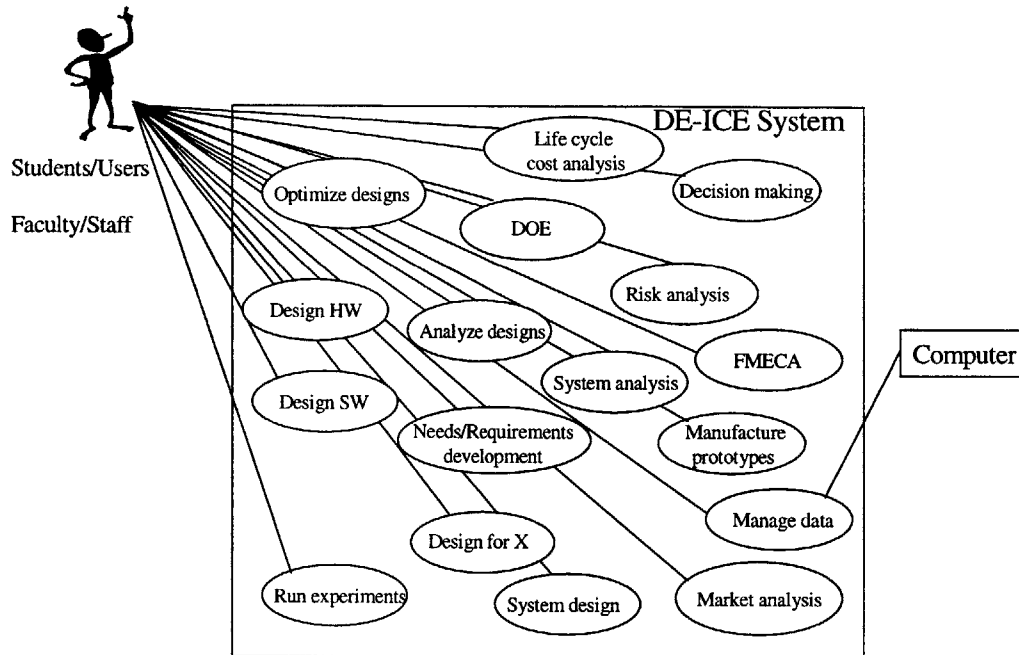


Figure 4.3: Design use case diagram

4.6 Chapter Summary

This chapter presented the technical requirements derived from the needs discussed in Chapter 3. They were classified as following:

1. Design and analysis support
2. Provide guidance throughout the design process
3. Planning and management of the design process
4. Experimentation support
5. Ability to operate on any platform

6. Distance collaborative support

7. Flexible system

8. Presentation and reporting support

Four main conflicts between technical requirements were presented and discussed. A use case study was finally performed to characterize and illustrate the way that users directly interact with the system.

Chapter 5

Design Center Architecture Study

5.1 Chapter Overview

Having identified the main technical requirements and the tasks to be achieved by the users, the next major milestone was to convert the requirements into an architecture. This chapter presents the necessary steps along with the tools used. A product matrix study was performed to identify and eventually select the hardware and software implementations that fulfill the technical requirements. Considering all the inputs collected, three architecture variants were designed by the team members. They were compared using Pugh's matrix (a technique used to visualize the evolution of a design concept from a baseline as it improves with time; see Pugh [21]) with the Baseline Architecture, an architecture previously designed by MIT staff. From the results of this comparison, a preferred architecture was selected.

5.2 Design Implementations Study Using a Product Matrix

To identify the design implementations that best support the most significant technical requirements, a detailed product matrix was developed. The score obtained by these implementations influenced the final architecture design. Using a product matrix allowed for traceability from design implementations to user needs.

The product matrix developed is similar to the QFD requirements matrix previously presented (see Figure 5.1 to Figure 5.4), except that the study is based on the most significant technical requirements instead of the needs. The selected technical requirements are displayed in the left column. Scores computed in the QFD matrix were used as a weight factor. Design implementations (hardware and software) are listed and categorized in the top row. The next step was to quantify how each of the design implementations satisfies

each of the technical requirements. The following relation factors were used: 9 = strong relation, 3 = moderate relation, 1 = weak relation, and zero (null) meant no relation at all. Like the QFD requirement matrix, the absolute scores were computed by multiplying the previous numbers by the weight factor of the technical requirement assessed, and by adding the scores obtained for each relationship for a particular design implementation. The absolute scores for each design implementation are displayed in the row named *TR Important Weight* in the product matrix, and the relative scores are displayed in the *Relative Weight* row.

The two highest rated implementations were DOME and Matlab. Matlab is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language. Matlab is very flexible, can be used for almost any kind of engineering calculation or simulation, and can be used on almost any platform. DOME enables the integration of complex tools in a relatively simple environment, operates on almost any platform, accommodates multiple simultaneous users, and can enable distance collaborative support. Many discipline-specific tools were highly rated, like CAD tools or orbit analysis tools (Satellite ToolKit). Guidance tools, like knowledge database or direct link to TAs, were considered very important. Systems engineering tools (like DeMAID for N2 matrices, QFD capturing tools or RDD-100) were highly rated as well.

Product Matrix																									
Design Implementations	Operating Systems				Printers				Simulation Tools				Modeling & Model Management Tools				Math Tools								
	TR Importance Weight	Windows 2000	Windows NT/6096	Mac	Unix	Color LaserJet	LaserJet	Plotter	Beamline	Digital Copy Machine	Copy Machine	Mathlab	MathCAD - System Build	Van Sim	Working Model	Grasdat	ADAM	Phenix Integration	MAUDE	PAU7	Access	Excel	DOE/	MathCAD	Mathematica
Design and analysis support	297				3	1	3		3	3	9	9	9	9	9	9	3	3	1	1	3	9	3	3	3
Provide guidance throughout design process	170																								
Planning and management of design process	149																								
Experimentation support	145									3				1							3				
Operate on any platform	143				1	1	1			1	9	1		1		1				1	1	9		1	9
Distance collaborative support	122	3	1	3	3			1	9	1			1					1		3		9			
Flexible system	98	3	3	1	3	3	3	3	3	3	9		1	1				1			3	3			
Presentation and reporting support	32	5	3	1		9	3	3	3	9	3		1	1							3		1		1
Design Priorities																									
TR Importance Weight	756	512	496	660	1616	830	1424	1100	2571	1546	5277	2816	2771	3068	2850	2816	891	1111	297	806	1859	5352	923	1034	2210
Relative Weight (%)	8	11	11	9	4	7	4	5	3	4	2	2	2	2	2	2	7	5	19	7	3	1	6	6	3
Architecture A		x				x	x	x	x	x		x	x								x	x			x
	756				1616	830	1424	1100	2571	5277		2771	3068								806	1859	5352		2210
Architecture B		x		x	x	x		x	x	x		x									x	x			x
	756		496	660	1616		1424	1100	2571	5277											806	1859			2210
Architecture C		x				x	x		x	x												x			
	756				1616	830		1100	1546	5277												1859			
Baseline Architecture			x			x		x		x											x	x		x	
			512			830		1100		5277											806	1859		923	
Architecture ABC'		x				x		x		x											x	x	x	x	
	756					830		2571		5277											806	1859	5352	923	

Figure 5.1: DE-ICE product matrix 1 of 4

Flexible System																	Presentation/Reporting									
3Ding (Video bridge)	3D email System	Palm Computers	Software Developer/Installer	Cell Phones w/ mobile	Laptops	Wearable Computers	Satellite Link	RFID support	DBL support	10 Mbps Ethernet 1 way	100 Mbps Ethernet two way	Wireless LAN	Tablets for all students who need it	Overhead Projectors	Video Projectors	Rolling furniture	Dimmable indirect Lighting	Portable Storage	Spoke on all wall switches	Blackboards/Whiteboards	File Charts	Lattes	MP Word	MP Power Point	Collaboration / Meeting Capture	
	1														1		1	1	1			1		1	340	
																									91	
	3																								83	
																									111	
	1							9	9		1	9						9	9		1	1			349	
	3	3	9	9	9	9	9	9	3	9	1	3		9				3						9	280	
	3	9	3	3	3	3		1	1	9	3	1	3	9	9	9	1	1	3	1	1	1	1		171	
				1							1		9	9	3	3		3	3	3	9	9	9	9	147	
1250	1545	1392	1424	1392	1392	1098	1098	1751	2483	1036	803	1673	1680	978	1275	882	857	1778	1974	386	826	529	1683			
5	4	4	4	4	4	5	5	4	3	6	7	4	4	6	5	7	7	4	3	14	7	11	4			
	x		x						x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Total score	% improv
	1545		1424						2483	1036	803	1673	1680	978	1275	882	857	1778	1974	386	826	529	1683	165405	61.43949699	
			x							x	x	x	x	x				x			x	x	x	Total score	% improv	
			1424							1036	803	1673	1680	978				1778				826	529	1683	81703	21.93554704
											x	x	x					x	x		x	x	x	Total score	% improv	
											803	1673	1680					1778	1974			826	529	1683	120136	48.90933809
	x								x			x	x	x	x			x			x	x		Total score	% improv	
	1545								2483			1673	1680	978	1275			1778				826	529	63781	0	
	x		x						x		x	x	x	x	x			x	x	x		x	x	Total score	% improv	
	1545		1424						2483		803	1673	1680	978	1275			857	1778	1974		826	529	118886	45.43315709	

Figure 5.4: DE-ICE product matrix 4 of 4

5.3 Architecture Variants

The next step was to create architecture variants from the results of the product matrix, and to rate and compare three variants. Each team member developed an architecture. Different views were used to describe the architectures: a software/IT conceptual view, an equipment block diagram and a physical view. Trade-offs between the best attributes of each variant and expected life-cycle cost were completed to make a final architecture. All architectures were compared to the Baseline Architecture.

5.3.1 Architecture A

Architecture A was the most complex and the most elaborate architecture variant developed. The three views illustrating this architecture are in Appendix F. A complete description of these three views can be found in Farnworth [17].

This architecture is composed of the following key architectural elements:

- Laptop PC for all users (Faculty and Students)
- Wireless network
- All software resident on laptop computers to reduce traffic on the wireless network
- CD-ROM for each student with all A/A software needed
- Liquid Crystal Display (LCD) second monitor for communication
- 100 Mbps Ethernet raceway
- DOME server to access the integrated modeling environment over the WWW
- NetMeeting/Collaborative Video Viewer (CVV)/Virtual Network Computing (VNC) for communication
- ViewStation video conferencing system
- Video bridge with software video switching
- Round Table Virtual Presence to improve distant meetings

- Electronic Writable Whiteboard to communicate sketches
- Software installation server
- Link to polymer rapid prototyping

5.3.2 Architecture B

Architecture B was developed by the author. This architecture emphasizes hardware flexibility and work space for students. It's the simplest of the three architectures developed, and it could easily be fully implemented for Fall 2000 classes since it does not use complex software under development like DOME.

5.3.2.1 Equipment Block Diagram

To satisfy the main technical requirements (see section 4.3), a set of hardware and software tools were implemented. The IT configuration can be seen in the equipment block diagram of Figure 5.5. A mix of PC and Mac laptop computers are used in this architecture. They are connected via a wireless network to three servers: a Unix, a PC and a Mac server. This configuration supports almost any kind of software since the three main platforms (PC, Mac and Unix) are available.

A docking station is provided at each work space. It includes a second LCD monitor, a small video camera and a NetPhone system to help in distance collaborative design (for example, one monitor could be used as the interface with the distant colleague while the other one could be used for drawing). The content of each monitor could be displayed on one of the two main screens in front of the room through the video projectors.

Architecture B Equipment Block Diagram

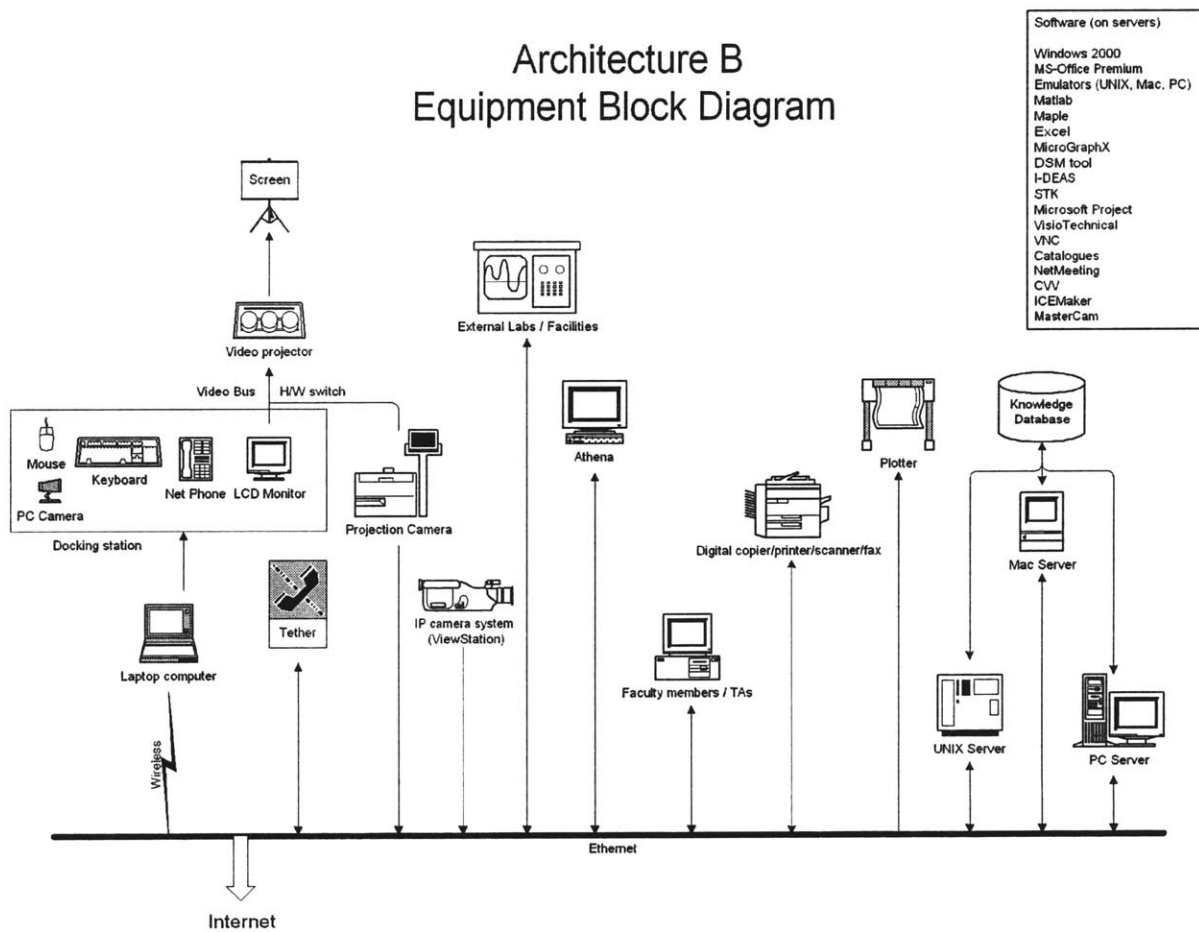


Figure 5.5: Architecture B equipment block diagram

An IP camera system (like the Polycom ViewStation) is used to send an overall picture of the main design room to distant sites through the Internet (high speed Internet). From the experience of the author, this system is simpler to use than the PictureTel system commonly used at MIT, but doesn't provide the same quality. Since other PictureTel facilities are available in the Learning Laboratory, an IP system that can be operated by the teaching assistant was judged satisfactory.

Links to Athena, external laboratories, external facilities, faculty members and teaching assistants are provided. A free tether connection is provided as well to enable the students to work with DE-ICE from their homes. A plotter and a digital copier (a machine that integrates photocopier, fax, scanner and printer) are available.

5.3.2.2 Physical View Description

The nominal room arrangement is presented in Figure 5.6 and is similar to the ones developed in industry (see sections 2.3.1 to 2.3.5). A lot of working space is provided for the students. The facility in which DE-ICE is implemented is divided into four rooms: one main design room and three small offices for team working. The room divisions are the same as the Baseline Architecture since the whole Learning Laboratory was already under construction when DE-ICE was introduced.

Interviews with faculty members (see section 3.3) showed that DE-ICE use includes different operational modes (like lecturing and performing designs in teams; see section 3.6 and Appendix D). Therefore, the ability to switch rapidly from one operational mode to another is very important. The physical architecture is composed of rollable furniture that enables a high level of flexibility. For instance, if a professor wants to interrupt a design session to provide theory or explanations to his students, the students just have to turn around and they have a desk in front or beside them to take notes. If the professor

wants to group the students in teams, the small desks between the work spaces can be used as work tables. The desks can be moved away very easily and joined to another desk if many students want to work together. The use of laptop computers and a wireless network enable the students to move the computers easily inside the facility in case they need it when they are working in teams. The large conference table at the middle could be used for long lectures, large group meetings, or simply as a work table if there are a lot of students in the class.

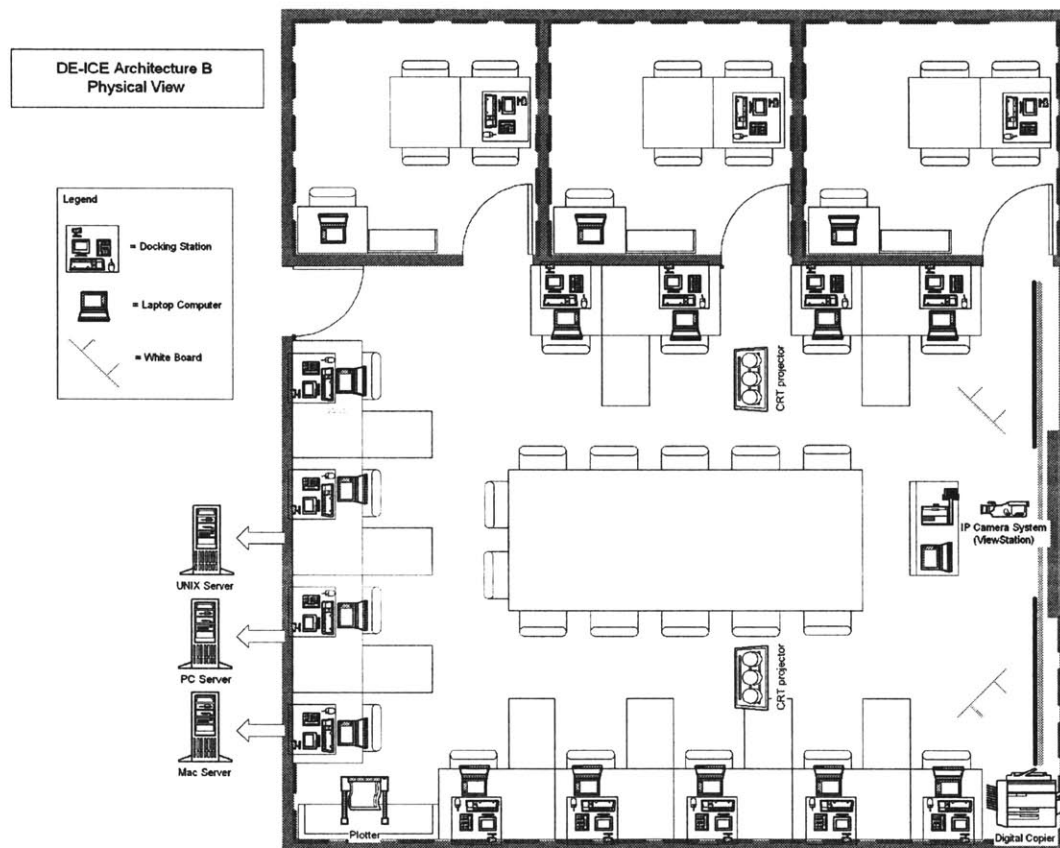


Figure 5.6: Architecture B physical view

The disposition of the projectors, the docking stations, the IP camera system and the digital copier can be seen in Figure 5.6. The three main servers are located in another room. A large moveable podium is available for the speaker, with a projection camera and

a computer linked to the network. One large whiteboard is fixed behind the projection screens, and two moveable whiteboards are available. Book shelves and other storage are installed in the laboratory as requested by faculty members (see section 3.3).

5.3.2.3 Architecture B software/IT conceptual view

The software architecture is illustrated in Figure 5.7. The main software tools are installed on the servers, and are remotely accessible to the users through the wireless network. Whenever a new software tool is installed on the system, or an old one is upgraded, only the servers have to be modified by the system administrator. The laptop computers use what is available on the servers. This should facilitate the maintenance of the system, and remove this responsibility from the students.

To communicate between Mac, Unix and PC computers, emulators (like Hummingbird Exceed) are used. For instance, if a student working on a Mac laptop wants to use CAD software available on the Unix server, he accesses the server through a Mac-Unix emulator and remotely runs the software from the emulator. Having the three different types of platform available in the laboratory makes it possible to easily upgrade the system with almost any software. Therefore, a minimum number of software packages is initially installed on the system. The architecture is designed to evolve with time, depending on the software needs for each particular class. Cost, CAD and knowledge databases are also available.

From the experience of existing industry design centers (see sections 2.3.1 to 2.3.3), and because of the intention of using mature technologies (see Table 1.1), Architecture B does not use a complex data integration environment like DOME. A simpler software

package developed by Caltech named ICEMaker (see section 2.3.6) is baselined. However, the architecture is designed to support DOME for future development and evolution.

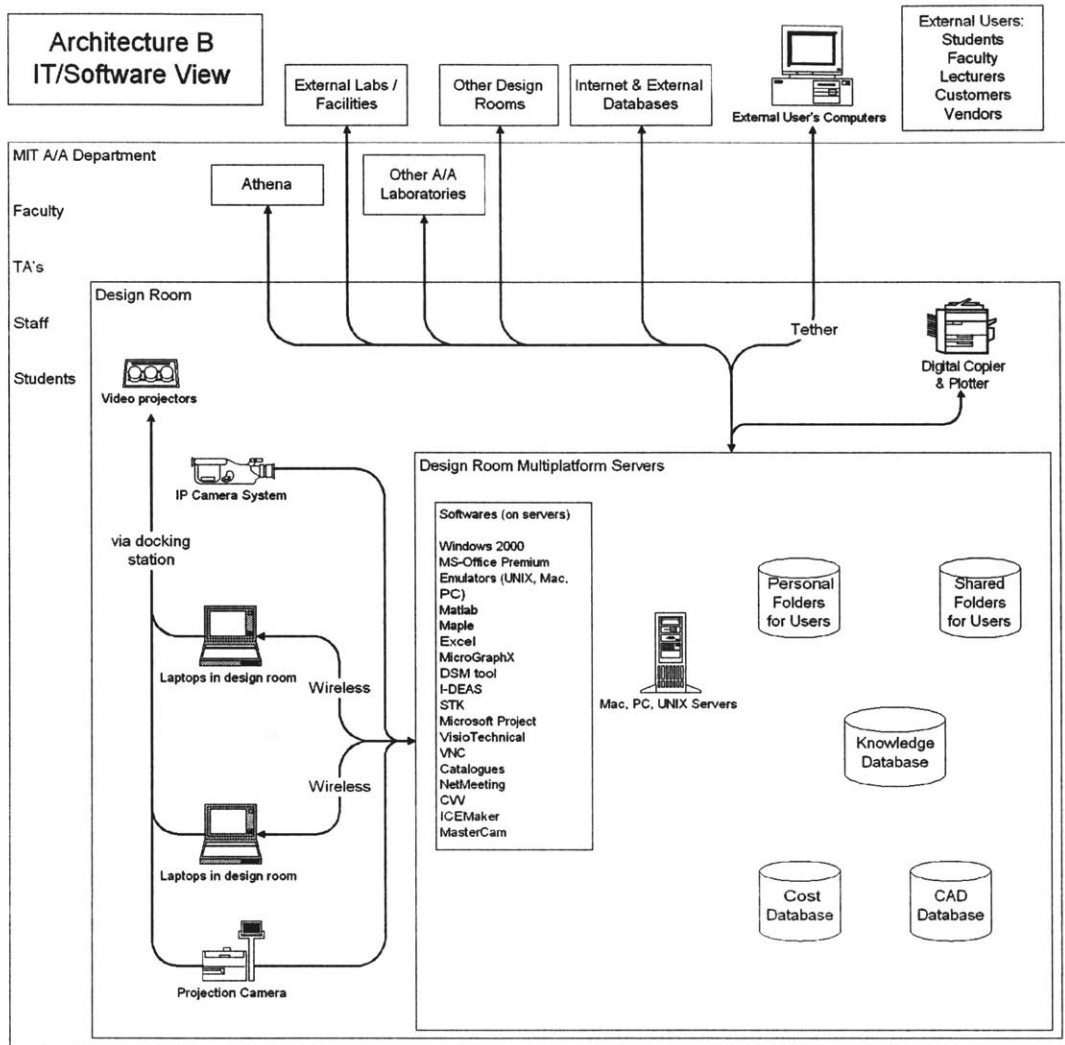


Figure 5.7: Architecture B software/IT conceptual view

5.3.3 Architecture C

Architecture C is the most software/IT-oriented architecture. It has been optimized for integrated concurrent engineering (ICE) sessions. The key architectural elements are:

- Windows 2000 workstations and servers

- Physical layout similar to the Baseline Architecture
- System and subsystem models integrated in Matlab
- Discipline-specific tools used offline
- System summary Excel spreadsheet for ICE sessions and external viewers
- Comprehensive guidance
- Workstations and external computers with equivalent design functionality

The software/IT conceptual view of this architecture is illustrated in Appendix F. A complete description can be found in Manka [22].

5.3.4 Baseline Architecture

The room arrangement of the Baseline Architecture was used for all the variants previously developed. The software/IT conceptual view and the physical view of this architecture can be found in Appendix F. Here are the key architectural elements:

- PCs with Windows NT
- Athena clusters in the laboratory
- All software resident on PCs
- 100 Mbps Ethernet to the desktops
- NetMeeting for communication
- Cornstarch rapid prototyping
- Links to external laboratories
- Links to faculty/TAs
- ProShare Video on each PC or PicTel type video conferencing (no decision has been made for the choice of the video conferencing system to be used)

More information about the Baseline Architecture can be found in Hallam [51].

5.4 Architectures Evaluations

The next step was to extract all the best elements from each architecture developed, and to integrate them in a final design. To evaluate and to compare the different architectures, both a QFD-type product matrix and Pugh's matrix were used. The Architecture B evaluation is described below. Evaluations of Architectures A and C can be found in Farnworth [17] and Manka [22].

5.4.1 Evaluation Through the Product Matrix

The evaluation of the architectures was achieved in the bottom section of the product matrix illustrated in the previous Figure 5.1 to Figure 5.4. For each architecture, all the design implementations used were checked, and the scores for each implementation were added. Final scores are illustrated on Figure 5.4.

This way of evaluating the architectures is tricky since an architecture having many implementations can obtain a high score even if the implementations are not compatible when used together. But it gives a good idea of weaknesses of an architecture. For example, the fact that Architecture B does not have any implementation for rapid prototyping or cost analysis is well reflected in the product matrix. These weaknesses have to be considered for the final architecture.

5.4.2 Evaluation Through Pugh's Matrix

Pugh's matrix was used to better capture the improvements in comparison with the Baseline Architecture for general criteria not assessed by the product matrix (like cost and functionality). The Baseline Architecture developed prior the introduction of DE-ICE is used as a datum. For each criteria, the architectures were compared with the datum using "+" for general improvement, "S" for similar and "-" for general deterioration (see Figure 5.8).

Comparing Architectural Variants




Architecture	Baseline	A	B	C	Preferred Characteristics	ABC'
Department Cost to Implement	Datum	-	-	-	Baseline: simple	S
Department Cost to Operate		-	-	S	C: no dual monitors, laptops, wireless network; common, off the shelf s/w	+
Design Functionality		+	S	+	A & C: integrated subsystem models, broad range of design tools, UV polymer for experimentation, brainstorming and design evolution via DOME in A	+
Communication Functionality		+	+	S	A: innovative remote presence, many modes avail., A & B: individual mobility	+
Flexibility		+	+	S	A & B: wireless LAN, room reconfigurable, B: OS mix, A: DOME & s/w distribution	+
Operational Modes		+	S	+	A & C: large systems, electronic whiteboard, knowledge database A, B & C : collaborative, distance learning	+
$\Sigma+$  $\Sigma-$  Σs 						ABC': Laptops supplied to all students as part of tuition, no maintenance or upgrade cost for laptops

Figure 5.8: Pugh's matrix for architecture evaluation

Architecture B scored “-” for cost to implement and cost to operate mainly because of the use of laptop computers with docking stations, and the use of a second LCD monitor and a wireless network. This high technology hardware has not been used extensively yet for teaching in the A/A department and some difficulties in the beginning are expected.

Due to the lack of an integrated environment like DOME, the Architecture B is comparable to the Baseline Architecture for design functionality (ICEMaker is used in the ICE sessions).

Communication functionality is improved using Architecture B because of the use of laptop computers and a second LCD monitor. It is possible for students to bring their laptop computers with them during team meetings. Efficiency of data communication and illustration sharing are improved. The use of a second LCD monitor makes it possible to communicate with distant colleagues on one monitor without losing work space on the other one.

Flexibility is enhanced because of the rollable furniture and the use of laptop computers. The room can be changed from a lecturing room to a design room in a few seconds. Laptop computers can be moved around the place easily and still be connected to the network via the wireless networking system. Tables can be moved quickly in case of team meetings.

Architecture B supports operational modes similar to the Baseline Architecture. No direct improvement was achieved due to the lack of maturity of the technology used to improve distant collaboration in design (like DOME, electronic whiteboards that are still relatively small and expensive, etc.). However, the architecture is designed to work with that technology once it is mature enough.

5.5 Architecture ABC'

The evaluation and comparison of each architecture enable the creation of the final one (called Architecture ABC') using some components of each variant. Since Architecture A obtained the highest score, the final architecture was based on its design. Figure 5.9 and Figure 5.10 illustrate the software/IT conceptual view and the equipment block diagram of Architecture ABC'.

From Pugh's matrix, it can be seen that the implementation and operation cost was the main difficulty of the three variants. To reduce the cost while maintaining the flexibility of Architecture A and B, it was decided that the laptop computers would be supplied by the students. The purchase of a laptop would be included in the tuition fee, removing this cost from the implementation cost of the architecture. Since most of the students are using a computer during their studies, this should not pose a problem. The A/A department would have to figure out what would be the best computer configuration for the students to have. A CD and a web site containing the tools needed by the students could be provided. Each

student would be responsible for the management of his/her own computer. Removing this responsibility from the faculty should help to reduce the operating/upgrading cost of the computers, assuming that a lot of instructions would be provided on the CD or through the web site. A new laptop could be provided on a regular basis to the students (like every two years for example), allowing them to follow the advance of the technology.

Part of the flexibility of Architecture B would be removed by requiring everybody in the laboratory to use PC laptop computers. Links to external Mac and Unix computers could be through the use of emulators. This should not be a problem for the students since it was demonstrated through the survey that they use more PC computers than Macintoshes (see section 3.4 and Appendix C). Faculty members, who are more reluctant to use PC computers and who are expected to access DE-ICE computers much less often than the students, still have access to DE-ICE through emulators.

The wireless network was abandoned. The main reason was its slow speed (in the order of 1 to 5 Mbps, compared to 100 Mbps for the standard network). This implementation will have to be reconsidered in the future when the wireless network reaches a satisfactory speed.

Some of the software tools used in Architecture A were not considered, but Architecture ABC' is designed to evolve with time depending on the needs of a specific class. Some hardware like the RTVP, the electronic whiteboard and the video bridge were removed, mainly for cost reasons.

It was decided that DOME would be implemented in the final architecture and its use will evolve with time when more integrating capabilities become mature enough to be used in classes. A cornstarch rapid prototyping machine is also available.

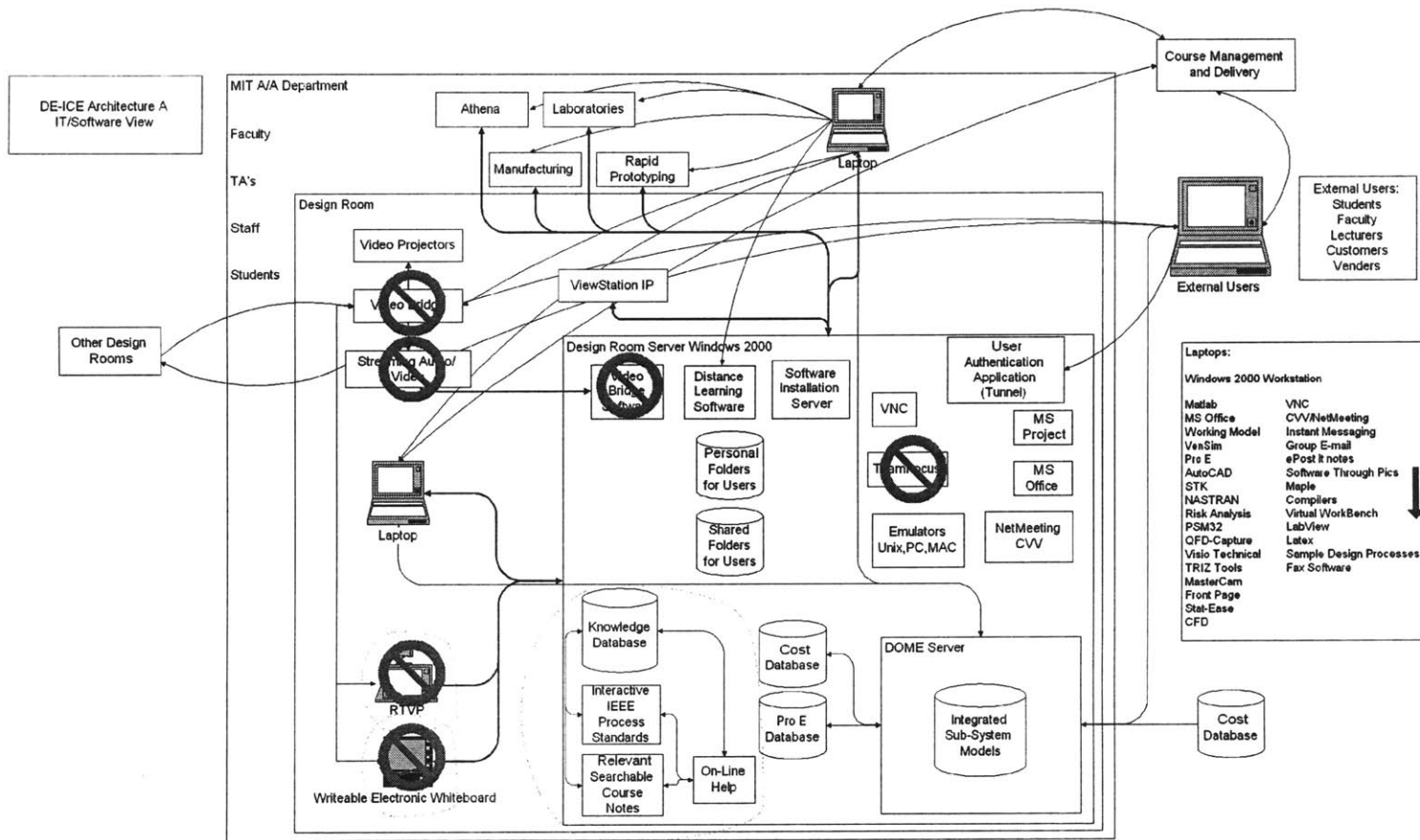


Figure 5.9: Architecture ABC' software/IT conceptual view (see Farnworth [17])

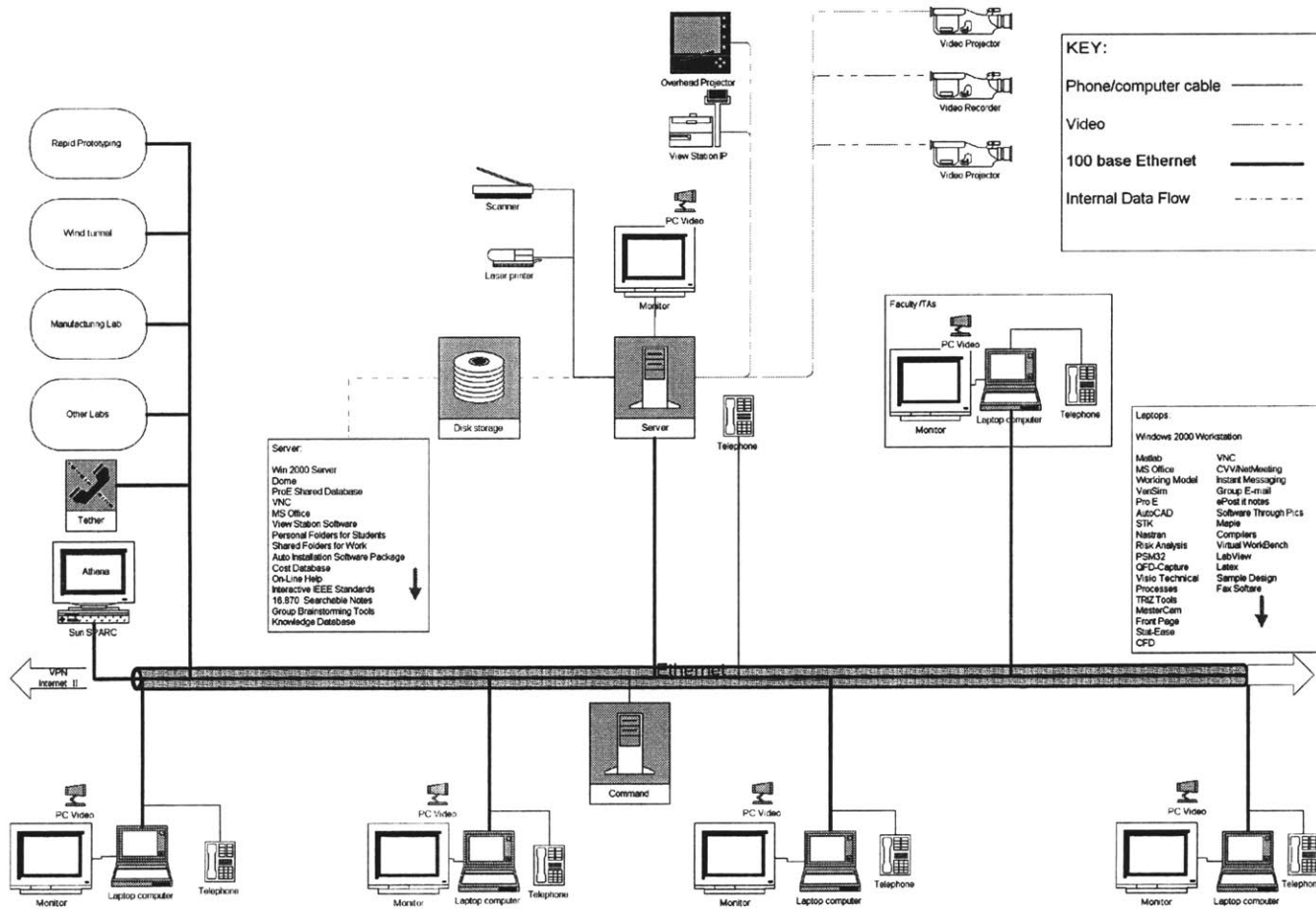


Figure 5.10: Architecture ABC' equipment block diagram (see Farnworth [17])

The product matrix (Figure 5.1 to Figure 5.4) shows the details of the design implementations of the final architecture. Pugh's matrix (Figure 5.8) illustrates the evaluation of Architecture ABC' compared to the Baseline Architecture. Significant improvements in almost every criteria can be seen.

5.6 Chapter Summary

This chapter presented the necessary steps completed to convert the requirements into an architecture. A product matrix study was performed to identify and eventually select the hardware and software implementations that fulfill the technical requirements. Considering all the inputs collected, three architecture variants were designed by the team members (Architecture A, B and C). Emphasis was placed on the architecture developed by the author (Architecture B). The three architectures were compared using Pugh's matrix with the Baseline Architecture, an architecture previously designed by MIT staff. From the results of this comparison, a final architecture (Architecture ABC') was developed using components of the three previous architectures. This architecture was recommended as the DE-ICE architecture.

The final architecture selected presents the following improvements over the Baseline Architecture: higher design and communication functionality, better flexibility, more operational modes supported and lower operation cost. Implementation cost is similar to the Baseline Architecture since the laptop computers used in the laboratory (PC computers using emulators to communicate with Macintosh and Unix platforms) would be supplied by the students. The author recommends strongly the presence of at least one Macintosh and one Unix computer in DE-ICE.

Chapter 6

Prototyping of a DE-ICE System Component

6.1 Chapter Overview

Having developed recommendations for the architecture of the design environment, the last goal of the project was to discover a key enabler of the system and to develop a component. This chapter presents the work of the team to discover the key enabler of the system. The prototype is then defined in light of the key enabler. The process used to design the prototype is presented. Emphasis is placed on the Work Manager since the author actively pursued the specification of this design element. Its functionality is illustrated and explained. Finally, a demonstration was achieved at the final design review of the project and is highlighted herein.

6.2 Prototype Selection

The first step was to discover a key enabler of the selected architecture for designing a prototype. The main ideas emerging from the elaboration of the architecture variants were:

1. The Round Table Virtual Presence (RTVP) system (see Farnworth [17])

The initial idea was to design the RTVP used in Architecture A's design. The RTVP promotes natural team presence for conferencing and consists of a triangular three-face monitor with a camera on the top and one microphone on each of its bottom corners (see illustration in Appendix G). The system enables automatic audio/video tracking of the speaker and is designed to be integrated with the IP camera system of Architecture ABC'. It also consists of modular units for easy reconfiguration.

2. The Electronic Writable Whiteboard (EWW) (see Farnworth [17])

The EWW consists of a large-format whiteboard capable of capturing and redistributing sketches, drawings and notes (see illustration in Appendix G). The data captured is available for use in any application. The EWW allows multi-user interaction and enables real-time communication and visualization for all team members through multiple distributed EWWs. Capabilities include the capture of interactive team decision processes and events, and the enhancement of team brainstorming. EWWs already exist on the market, like the *mimio* system, but some lack complete functionality.

3. Integrated On-Line Teaching Assistant (see Farnworth [17])

The integrated On-Line Teaching Assistant (On-Line TA) is envisioned as a graphic and interactive system focused on the product development process (PDP), providing access to relevant course notes, sample PDPs, Systems Engineering Management Plan (SEMP) templates, external resources, and other relevant information (see illustration in Appendix G). The On-Line TA is capable of acquiring knowledge from current projects and retaining this knowledge in a knowledge database for future reference. An automatic capture of student's design objects into this knowledge database is provided. Drag and drop of objects is also supported. The interface with the user could be fully customized.

4. Distance Learning Software

The last prototype idea was to design a software capability that would improve distance learning education by integrating audio, video and some useful applications like a video image of each distance student, a warning feature when somebody has a question to ask, a system that links in real-time student's notes taken electronically with the presentation slides, etc. Actual systems have limited functionality and quality (see Williamson [10] and Nolet [31]). The Microsoft Tele-Presence

Presentation system (TELEP) and the Microsoft Research Annotation Service (MRAS), both systems in development at Microsoft, have aspects of the desired capability.

To select the basis for the prototype, a matrix very similar to a product matrix was developed (see Figure 6.1). Relations between each technical requirement and each prototype concept were identified. Final scores clearly identify the On-Line TA as being the most relevant near-term prototype concept for the DE-ICE system.

Technical Requirements	Prototype Concepts				
	TR Importance Weight	RTVP	e-Whiteboard	On-Line TA	Distance Learning Software
Design and analysis support	297		1	9	
Provide guidance throughout design process	170			9	
Planning and management of design process	149			3	
Experimentation support	145			1	
Operate on any platform	143	9	9	9	9
Distance collaborative support	122	9	9	1	3
Flexible system	98	3	1	1	9
Presentation and reporting support	32	1	9	1	1
Design priorities				★	
TR Importance Weight		2711	3068	6334	2567
Relative Weight (%)		3	3	1	3

Figure 6.1: Prototype selection matrix

The On-Line TA is the only prototype idea that addresses the most important technical requirements. The On-Line TA is a key element of DE-ICE, providing guidance to users who lack an extensive industry background, providing information on when and how to accomplish design goals, and helping users tailor DE-ICE resources to their needs. The On-Line TA integrates multiple components into a highly usable system, performs certain project manager functions, and automatically links student work. Using the On-Line TA, the student work entities could be used as measurable outcomes for grading and accreditation support to the faculty.

6.3 On-Line TA Use Case Diagram

A use case diagram (see Figure 6.2) was prepared to characterize and illustrate the way in which users directly interact with the On-Line TA. From this diagram, four main users were identified: students, faculty, teaching assistants and staff members.

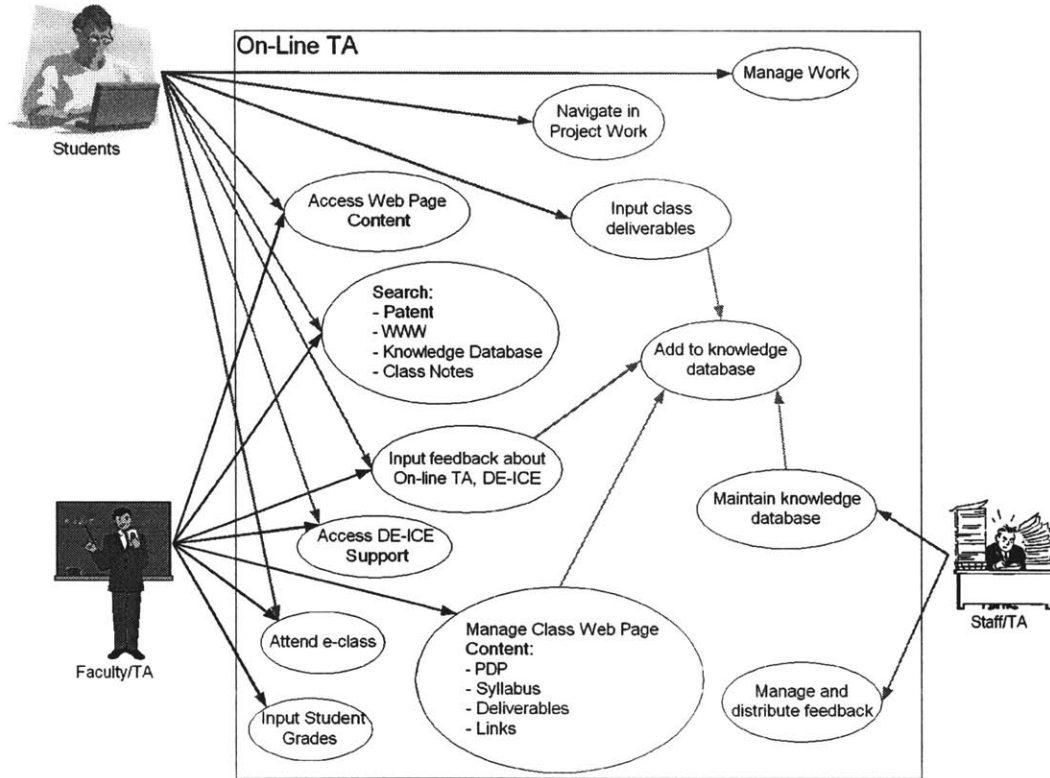


Figure 6.2: On-Line TA use case diagram

This use case diagram was used to determine the different components of the On-Line TA.

6.4 The On-Line TA Prototype Breakdown

Considering the use case diagram in Figure 6.2, the decision to decompose the On-Line TA concept in the following four major components was made: a web-based front end interface, a front end for On-Line TA authoring, an interface to allow students to interact

with their work and a project navigator. The system should be easy and fast for faculty to create and customize for their specific needs, and should be fun for all users. Loading time for the On-Line TA should be as fast as possible. Appendix G illustrates how the work was split between the team members, and which components were prioritized.

6.4.1 Front End for User Interface

The user interface is a graphic web-based (platform and OS independent) application that integrates (see Farnworth [17]):

1. A general PDP, like NASA Program/Project Life Cycle, Ulrich and Eppinger, etc.
2. Tools to create and analyze a PDP for a class/project, like Design Structure Matrix (DSM), or Program Evaluation & Review Technique (PERT)
3. On-Line TA/DE-ICE help
4. Direct link to faculty and TAs through the Internet
5. Feedback on the DE-ICE system/On-Line TA/classes
6. Advanced search tools to gather information from multiple resources (i.e. patent searches, knowledge database, lessons learned, course notes, SEMP, etc.)
7. Tool support in the DE-ICE system
8. Class materials that contain deliverables and timing
9. Class deliverables submission through the On-Line TA
10. Electronic classes and lectures

Students logged into the system have full access to resources on DE-ICE. Guests should have the capability to log in and utilize some of the functionality. The log-in process can be used to:

11. Generate the class customized PDP for a particular user
12. Automatically update only the changed information from the authoring tools

13. Collect user information and access time/date
14. Tag objects
15. Capture objects
16. Collect electronic class attendance information
17. Collect On-Line TA/DE-ICE usage information

6.4.2 Front End Authoring Tool for Customizing the On-Line TA

Because the On-Line TA will quickly become a very complex network of web pages, an authoring tool is needed to custom script the web pages and to reduce the time requirement placed on faculty when creating/tailoring the On-Line TA for a specific class. The authoring tool is needed to perform the following (see Farnworth [17]):

1. Define a top-level PDP for the class
2. Customize an existing top-level PDP
3. Highlight PDP phase tasks that contain deliverables
4. Lay out deliverable schedule
5. Define PDP in phase tasks
6. Insert content to be displayed for top-level and in phase tasks
7. Define class goals and objectives
8. Insert class syllabus
9. Define class notes
10. Plug-in new tools for class use
11. Build the customized On-Line TA web pages
12. Automatically verify page links and view page layouts as thumbnails
13. Automatically record when changes are made to update the On-Line TA

Time required of the faculty to interface with the authoring tool should be less than 1/10 of the normal class preparation time, not including preparation of the actual content. This is a design constraint identified by the team.

6.4.3 Work Manager

The Work Manager consists of a multi-platform/multi-user interface acting between the students and the knowledge database. It enables students to manage their work, whether they are working in a team or by themselves, on campus, from home or even through the Internet. Direct interactions with the Work Manager are illustrated in the use case diagram of Figure 6.3.

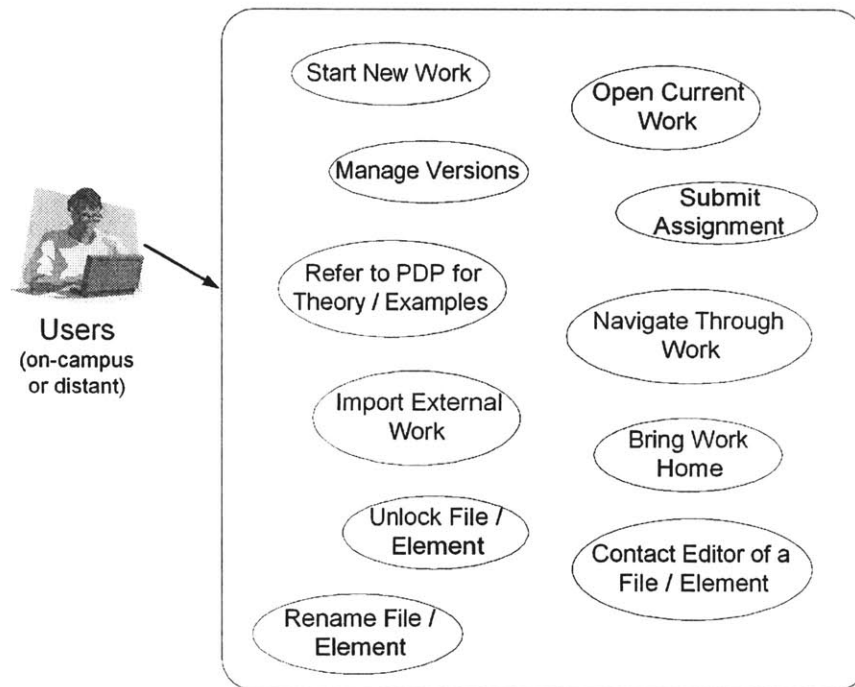


Figure 6.3: Work Manager use case diagram

Some functions, like *Manage Versions*, *Import External Work*, *Bring Work Home*, *Unlock File/Element*, *Contact Editor of a File/Element* and *Rename File/Element* are an

expansion of the function *Manage Work* of the use case diagram in Figure 6.2. They were all judged necessary from the experience of the author to support design (the most important technical requirement retained).

The Work Manager is accessible to the users from the Front End for User interface. Because of interactions with multiple software tools, the Work Manager is not integrated in each software tool. Instead, the Work Manager is running in the background through an applet loaded when somebody logs in. The Work Manager monitors the files directly on the disk where they are stored (knowledge database) and warns the user when a file has been modified (with a “*” beside the name on the Work Manager interface). The student can make a new version of that file, project element, or even of the entire project database.

6.4.4 Project Navigator

The Project Navigator provides a roadmap allowing the visualization of the design process work in progress (with inputs and outputs). The Project Navigator is accessible from the Front End for User interface and the Work Manager. Each waypoint in the process is associated with objects (presentations, design documents, prototypes). The relationship between these objects is illustrated. Typically, the work is decomposed into different objects and aggregated again at the project milestones and reviews. The Project Navigator is used as (see Manka [22]):

1. A record

The working team uses the Project Navigator to map their project. The Project Navigator archives the project’s *whats* and *hows*. It provides traceability of the design decisions.

2. A guide

Professors and TAs use the Project Navigator to build the roadmap of the project deliverables and milestones. The Project Navigator suggests the steps to be used. It includes data like descriptions, scheduling, number of people assigned, etc.

3. Both

Students build off the templates as they go.

6.5 Work Manager Functionality

Because of the software nature of the On-Line TA, its complexity, and the limited time to design it, the team decided to develop a prototype of 1) sets of functional flow diagrams (FFD) illustrating the functionality and 2) a mock-up demonstration showing how the software behaves. Once this functionality is clearly defined and demonstrated, additional development will be possible in the future.

This section presents the functionality of the On-Line TA (mainly the Work Manager) through different FFDs. Emphasis is placed on functionality developed by the author of this thesis. Figure 6.4 shows the On-Line TA top level FFD and gives an overview of the functions performed by the whole system. The author developed the function *1.0 Provide Log-In Service*, *7.0 Access DE-ICE Tools*, *7.8 Submit Homework* and other sub-functions (see Figure 6.8 to Figure 6.12).

The first function developed was *1.0 Provide Log-In Service* (see Figure 6.5). This function is used anytime somebody wants to log on the On-Line TA. The first task accomplished by the software is to identify the user. Once the user is identified, depending on the type of user, appropriate access permissions are granted, preventing unauthorized users from modifying any system database. After proper identification, some tasks are accomplished simultaneously. The most important ones are 1) asking the users the class/project they would like to work on so that the correct PDP for that project is loaded and 2) launch-

ing the Work Manager applet used to monitor file attributes in the knowledge database. The last task is to personalize the web page (item 1.6) depending on the user information collected

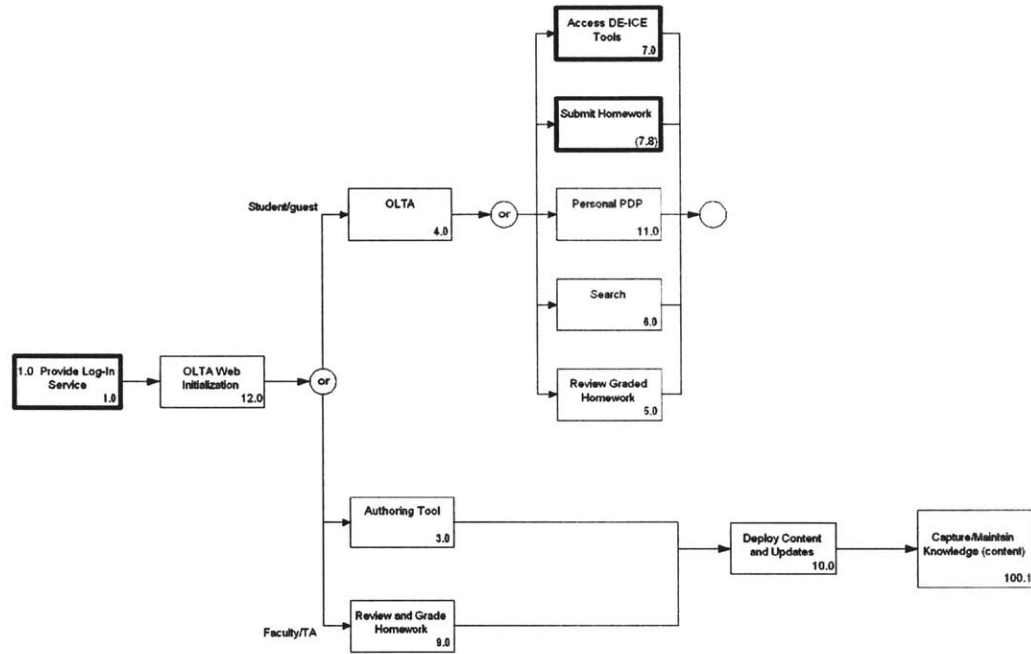


Figure 6.4: On-Line TA top level FFD

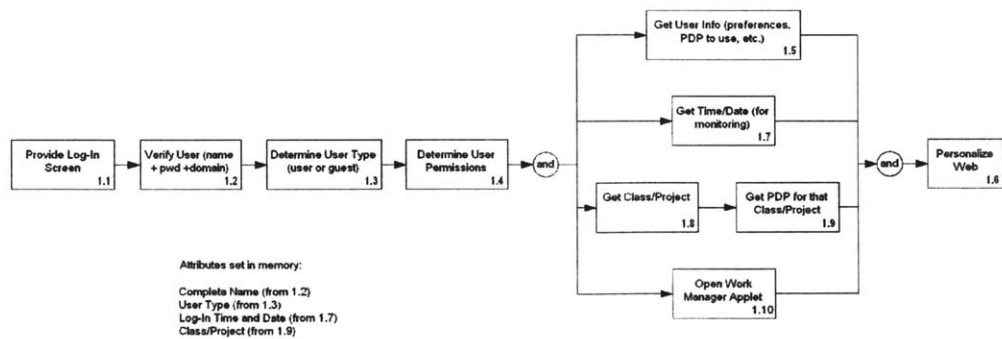


Figure 6.5: Log-in service FFD

The main application developed by the author was the Work Manager defined in section 6.4.3. The interface is illustrated in Figure 6.6. Each of the functions accomplished by the interface are illustrated using a FFD. The interface is Java-based for multi-platform and access through the Internet.

The first three columns on the left of the Work Manager main window are used for hierarchical classification, the same way files and directories are related to each other in a conventional operating system. The first column (*Project*), is the highest entity in the hierarchy. The second one (*Project Element*) contains the subdirectories associated with the project. The *Filename* column lists the name of the files associated with each project element. Pale file names represent the ones that are viewable by multiple users. The others are viewable only by the user who logged on. A simple click on the name of a file brings a box asking for the correct version of the file to open. Multiple versions of the file would be stored in the system (knowledge database) and would be accessible anytime.

The fourth column indicates the tool used to make a particular file. A click on a tool name creates a new file and associates by default the file created to the corresponding *Project Element*.

The fifth column shows the corresponding section in the PDP linked to the file. If a user wants to have more information and theory, a click on the link brings him back to the PDP where he could find information related to the file. For example, if a file is a QFD matrix, the link connects the user to the section in the PDP giving theory and examples about QFD matrices.

The next three columns are related to version management and submitting work. They indicate when was the last time a file was modified, what is the latest version number, and what is the version number of the latest version of the file submitted to the professor. The

Last Modified column is a real-time status given by the Work Manager applet launched during the log-in process.

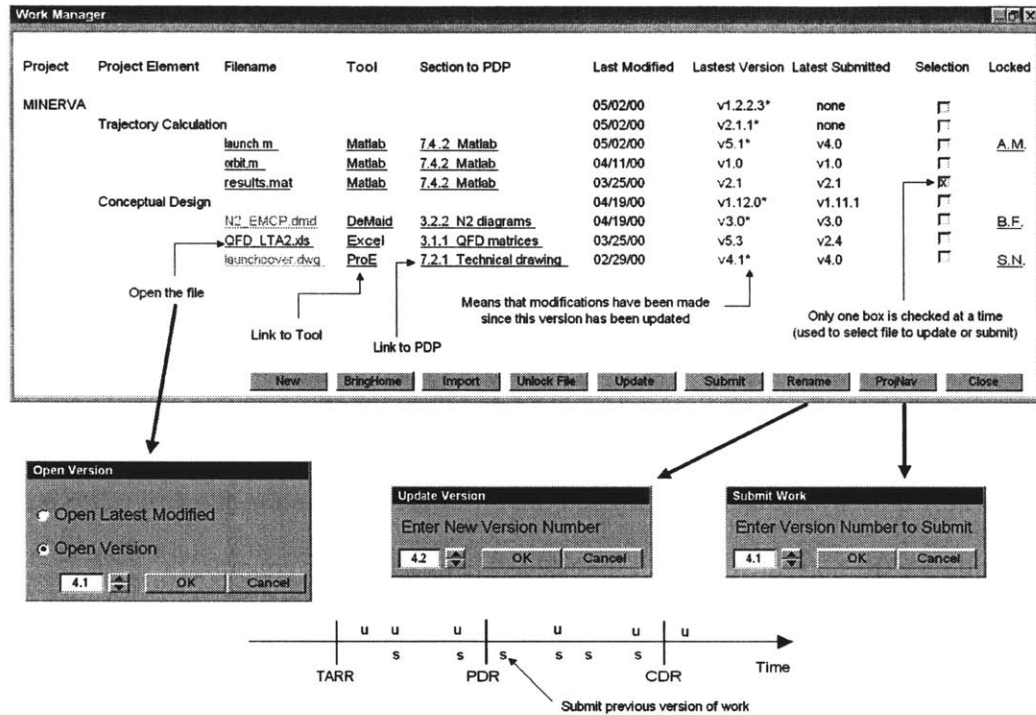


Figure 6.6: The On-Line TA Work Manager interface

The *Selection* column is used to select one or multiple entities (files, project elements or projects). The *Locked* column is used to specify locked entities and contact whoever locked an entity. More information about the buttons' functionality is provided below using FFDs.

The timeline in Figure 6.6 represents a possible scenario of file updating and file submission for a student. In this example, the project is divided into three milestones (Trade Analysis and Requirements Review, Preliminary Design Review, Critical Design Review). The student can update the version number of a file as often as he wishes. Each time a new version number of a file is registered, a copy of the file as it is during the regis-

tration process is automatically kept in memory for future reference. Each time a new version of a file is submitted before the deadline to a professor, the previous version is automatically deleted from the professor's directory. A deleted file from the professor's directory is still kept in the main memory of the system. The student can submit any version of the work, not necessarily the latest, as is shown in the example.

Figure 6.7 shows the top-level FFD for the Work Manager. After a click of the user on the Work Manager link from the On-Line TA Front End (see Farnworth [17]), some variables are loaded in memory to be used later (the position of the user in the PDP when he clicks on the Work Manager link, and the project name acquired during the log-in process). Then, the Work Manager interface is loaded and a lot of functions can be accessed by the student (see Figure 6.7). Appendix H contains the interfaces interacting with these functions and developed for the purpose of the demonstration.

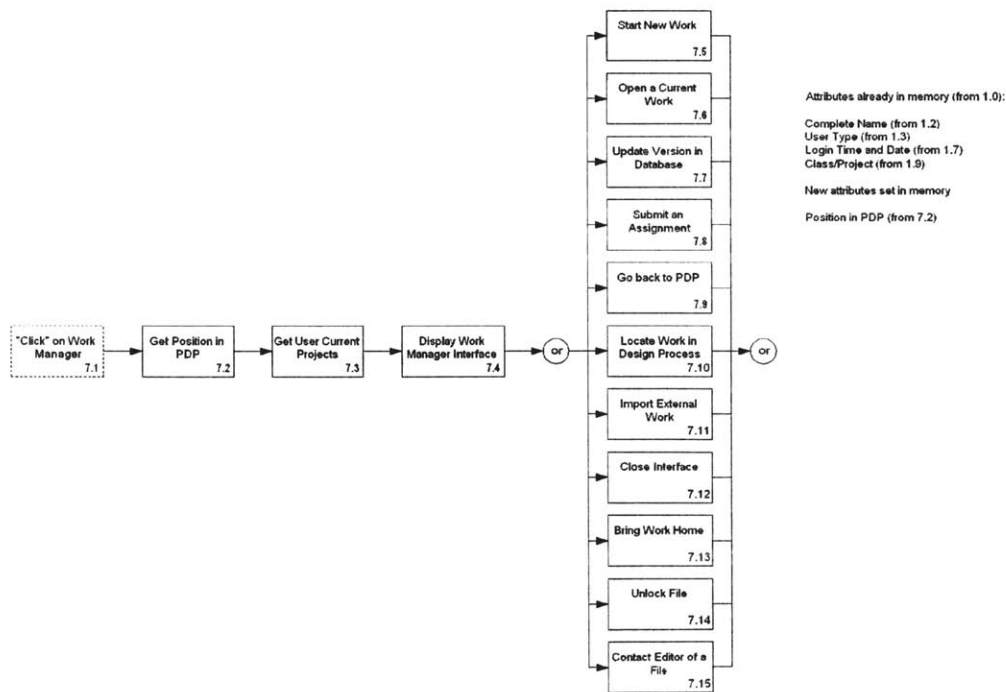


Figure 6.7: Work Manager top level FFD

The first function developed was the *7.5 Start New Work* function (see Figure 6.8). This function is activated either by a click on the *New* button or on the tool name as explained previously. The time and date are the first variables set in memory. If the user clicks on the *New* button instead of the tool name link, he has to choose the name of the tool he wants to use (Excel, PowerPoint, Autocad, etc.). After the tool name is acquired, an interface (see Appendix H) appears with default values gathered in previous functions (see the attributes' description in Figure 6.8). Through the use of the interface, new attributes are set in the memory, like the multi-user permission that decides to whom the file is accessible. Once all the necessary information is gathered, an instance is created in the knowledge database, the tool is launched and the Work Manager main interface is updated. The file is automatically locked until the user decides to manually unlock it. While the file is locked, the name of the person who owns the write permission is displayed in the Work Manager main interface, and the file is still accessible to other users in the *read-only* mode.

The *7.6 Open Current Work* function is simpler since the entity is already created and monitored in the knowledge database. The function is activated when somebody clicks on the name of a file. The program first asks the version number to open (see interface in Appendix H). If the file is unlocked, the program asks the mode of editing (with write permission or read-only). If the user selects to open the file with writing permissions, the program locks it to avoid having more than one user at a time trying to edit the same file. If the file is already locked, the program automatically opens it in read-only mode. When the locking process is achieved, the program is launched (through an emulator if it is installed on another platform) and the file is loaded.

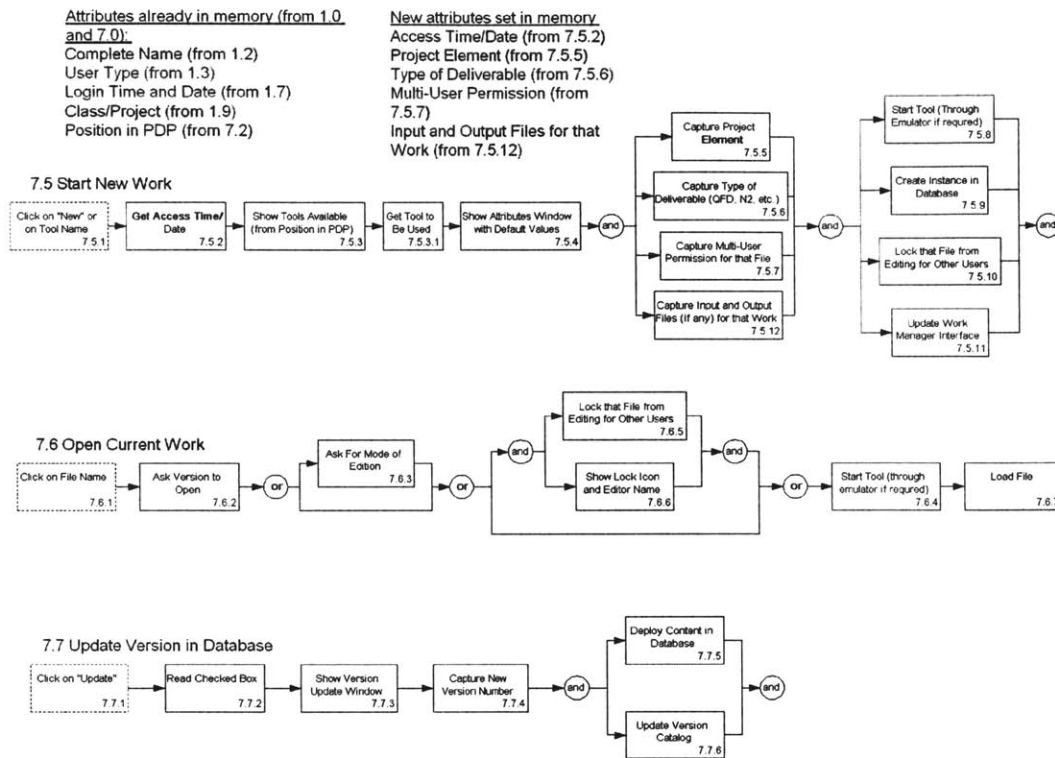


Figure 6.8: Work Manager detailed FFD 1 of 5

When somebody is editing a file (any file running on any program and any platform) and performs the *Save* function in the program, the *Latest Modified* attribute for that particular file is updated. This is true for any operating system. This attribute is monitored by the Work Manager applet launched during the log-in process. When the applet monitors that the file has been modified after the latest version was made, a “*” appears beside the latest version number in the *Latest Modified* column for that file and the higher level entities (project element and project; see Figure 6.6).

When the editor of an entity wants to make another version of it, he first selects the file by checking the appropriate box and clicking on *Update*. The program reads the box checked and displays the Update interface (see Appendix H). It records the new version

number entered by the user, stores the file in the database and updates the Work Manager main interface (see Figure 6.8).

The version numbering works as follows: for a minor change on the previous file, the user is encouraged to increment only the decimal digit of the version number. For a major change, incrementing the unit digit is suggested. Each time the version number on a file is updated, the version number of every higher level entity is automatically updated, but not the one of the lower level entities. For example, updating *launchcover.dwg* in Figure 6.6 from version 4.1 to 4.2 will update *Conceptual Design* project element from 1.12.0 to 1.12.1 and *Minerva* project from 1.2.2.3 to 1.2.2.4. Updating *launchcover.dwg* from version 4.1 to 5.0 will update *Conceptual Design* project element from 1.12.0 to 1.13.0 and *Minerva* project from 1.2.2.3 to 1.2.3.0. Updating *Conceptual Design* project element from 1.12.0 to 2.0.0 will update *Minerva* project from 1.2.2.3 to 1.3.0.0, but will not change the version number of *launchcover.dwg*. Updating *Minerva* project from 1.2.2.3 to 2.0.0.0 will not change anything else. High-level entities cannot be updated if all the lower level entities have not been updated before.

Another main function of the Work Manager is *7.8 Submit an Assignment* (see Figure 6.9). It is activated after a user checks an entity to submit and presses the *Submit* button.

The first step the program achieves is to compare the time and date of the submission with the deadline entered by the professor for that entity (this feature is not fully implemented in the current version of the Work Manager). If the deadline is past, the professor has the choice of accepting the work or rejecting it. If he accepts it or if the deadline is not past yet, the submission interface (see Appendix H) appears, allowing the user to choose the version number of the entity to submit. The program erases the previous version of the file in the professor's directory (if any) and copies the desired version of the file in the same directory.

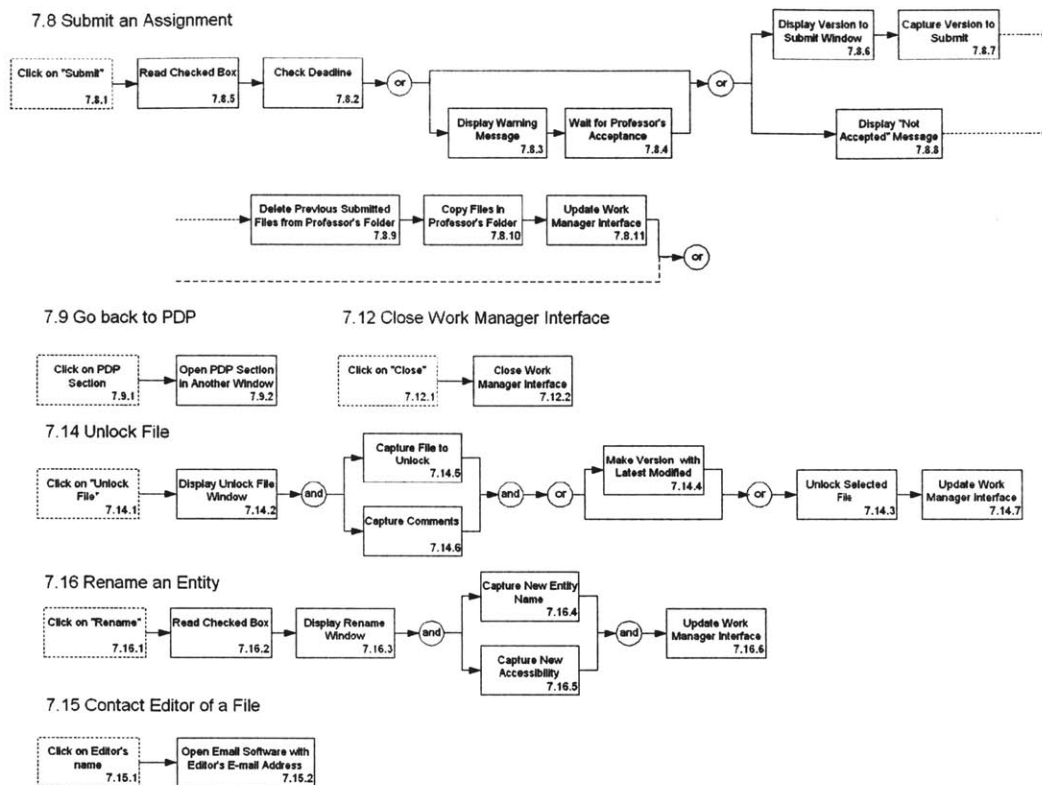


Figure 6.9: Work Manager detailed FFD 2 of 5

Function *7.14 Unlock File* is used when somebody wants to return the editing permission of an entity to the system so that it will be available to other users. It automatically makes a new version of the entity and all the lower level entities not already updated if there is a "*" associated with the entity.

Function *7.16 Rename an Entity* is used to change the name of an entity or to change its multi-user accessibility. When the name of an entity is changed, the update is automatically propagated in the previous versions of that entity. The other functions of Figure 6.9 are very simple and so the FFD description is sufficient.

Function 7.10 *Locate Work in Design Process* (Project Navigator) of Figure 6.10 has been developed by Manka [22]. This function uses attributes gathered when creating a new entity in the Work Manager.

Function 7.13 *Bring Work Home* illustrated the process followed by the program when a user wants to work on an entity somewhere the Internet is not accessible. The function is activated when the user checks the entity and presses the *BringHome* button (see Figure 6.11). An interface appears (see Appendix H) to ask the user 1) whether the intention is to modify the entity or not, 2) the version number of the entity to bring home and 3) the location where the file has to be copied. If the user wants to edit the entity at home, the program locks this entity. The user can take the entity for this purpose only if no other user already locked the file.

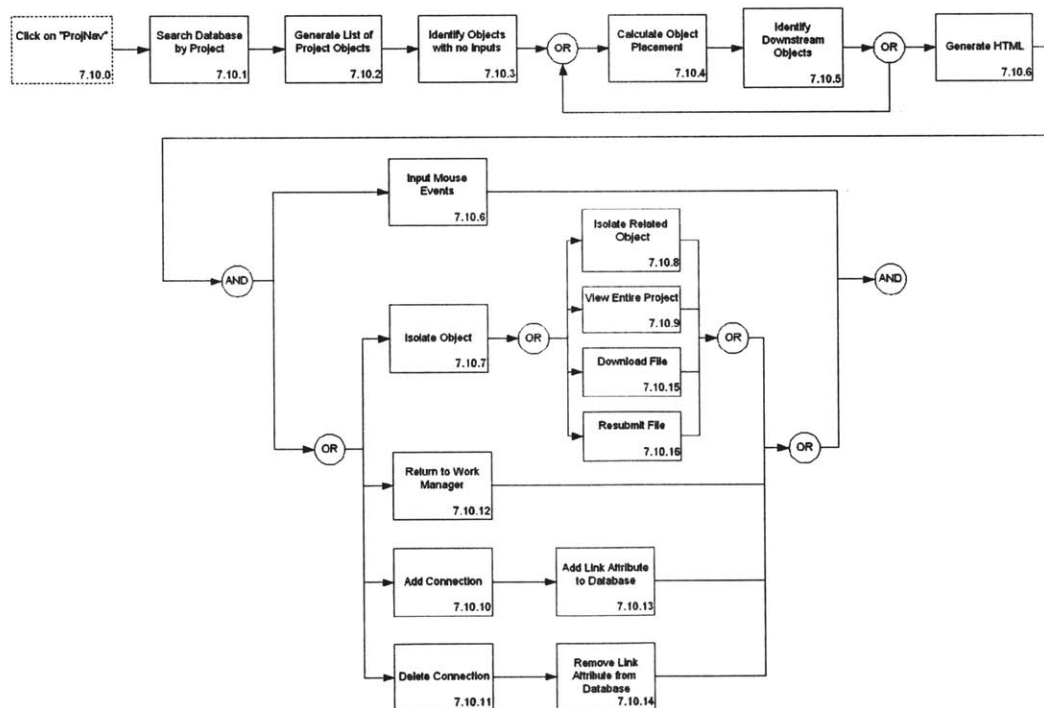


Figure 6.10: Work Manager detailed FFD 3 of 5 (see Manka [22])

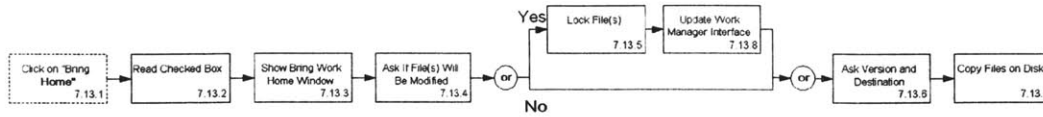


Figure 6.11: Work Manager detailed FFD 4 of 5

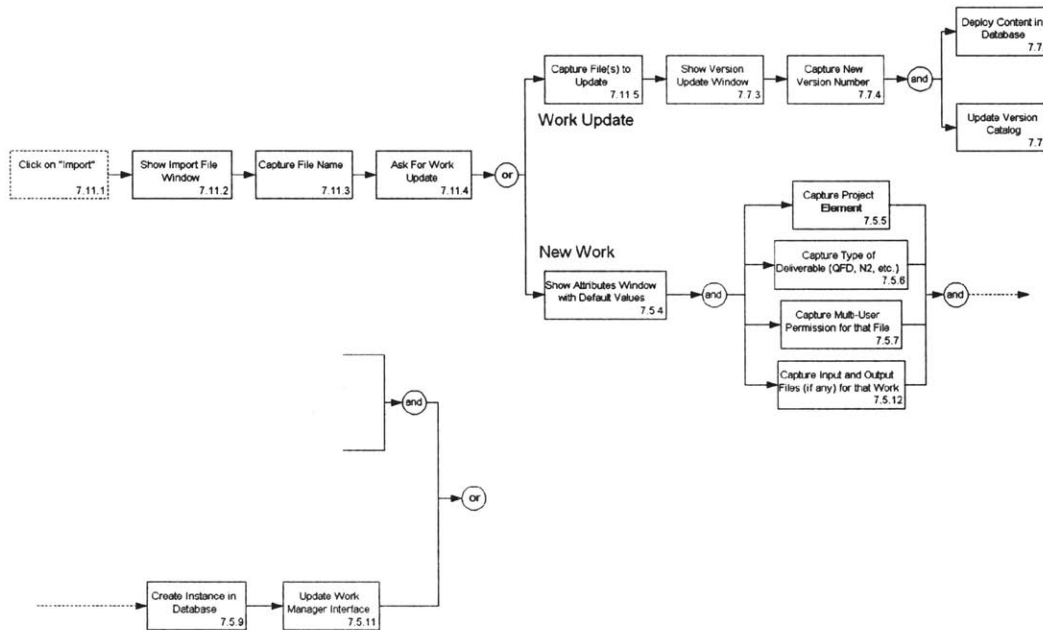


Figure 6.12: Work Manager detailed FFD 5 of 5

The last important function is *7.11 Import External Work* (see Figure 6.12). This function is used when somebody checked a file out and wants to get it back into the system, or to input outside-initiated entities into the system. Again, when a user clicks on the *Import* button, an interface appears (see Appendix H). The interface is used to indicate the location of the file to import. The user has to indicate if the work to import is new or if it is an updated version of an entity already in the system. If it is new, the same attributes as the

ones associated with the *7.5 Start New Work* function are input by the user and the entity is created in the system. If it is an updated version of an entity already in the system, the program 1) asks for the new version number, 2) copies the file in the database and 3) updates the Work Manager main interface.

6.6 Demonstration

As mentioned previously, the prototype consists of sets of FFDs illustrating the functionality and a mock-up demonstration showing how the software would be used. The suggestion of doing a mock-up demonstration came from experts consulted (Microsoft and Wertheim Inc.).

The first idea was to integrate existing software into the On-Line TA so that a real working prototype could be implemented. Research and consultations were performed to identify the appropriate software tools. The main tools considered were:

- The Hummingbird Enterprise Information Portal (EIP) including PCDocs Fulcrum and CyberDocs
- Rational ClearCase
- Microsoft Site Server

The first two are used mainly for e-Commerce applications, while the third one is a tool facilitating the creation of sophisticated World Wide Web-based educational environments (see Goldberg [23]). Microsoft Site Server is used to create and manage sophisticated web pages. None of these packages integrated all of the functionality of the On-Line TA and the Work Manager. Investigations were made to see if it is possible to tailor PCDocs Fulcrum to perform the Work Manager functions. This approach was finally rejected because of the associated high cost and risk. Even having PCDocs Fulcrum tailored to the Work Manager would have allowed demonstrating only limited functionality

of the Work Manager. The creation of the Work Manager using Microsoft Site Server was considered as well. This idea was abandoned due to the complexity of Microsoft Site Server. For all these reasons and the project time constraints, it was decided that a mock-up demonstration would be the best option.

The demonstration was performed during the final design review of the project. It consisted of a set of web interfaces, pictures and documents linked together. The web interfaces for the Work Manager are displayed in Appendix H with the sequence of actions taken during the demonstration. The demonstration showed how the software would behave when the user inputs certain commands. The web interfaces were designed to be very representative of the real software.

6.7 Chapter Summary

This chapter presented the work performed by the team to discover the key enabler of the system, the integrated On-Line Teaching Assistant (On-Line TA), a graphic and interactive system focused on the product development process (PDP), providing access to relevant course notes, sample PDPs, Systems Engineering Management Plan (SEMP) templates, external resources, and other relevant information. The prototype was then defined in light of the key enabler and was divided into four main elements: a Front End for User interface, a Front End Authoring Tool for Customizing the On-Line TA, a Work Manager and a Project Navigator. Emphasis was placed on the Work Manager developed by the author. The Work Manager is a multi-platform/multi-user interface accessible on campus, from home or even through the Internet to help the students to manage their work. The Work Manager is not integrated in each software tool, but runs in the background and monitors students' files directly on the disk where they are stored (knowledge database). Finally, the demonstration achieved at the final design review of the project was highlighted.

Chapter 7

System Implementation Plan and Recommendations for Future Work

7.1 Chapter Overview

In order to move forward with the project, the A/A department needs estimates of time, costs and impacts involved by the implementation of the On-Line TA. This chapter presents future work recommended for the On-Line TA concept through a system implementation plan. The level of completion of the On-Line TA is quantified to help identify and select future design options. Time and cost estimates for each option are provided, as well as a suggested schedule. Then, the work needed to be done in order to implement the whole DE-ICE system as designed by the team is specified. Finally, recommendations and conclusions on the overall project are made.

7.2 Level of Completion of the Prototype

Software design experts from Integrated Information Systems (IIS) were consulted to evaluate the On-Line TA's level of completion. The level of completion estimate is based on the design process presented in Figure 7.1. A description of each phase of this design process follows (see Appendix I):

1. Conceive

This phase is used to define the project scope. Critical issues are assessed and experts explore how to resolve them. An overall vision of the project is provided and strategic objectives are set.

2. Explore

Requirements are established in this phase, as well as preliminary project plans and metrics. A technical feasibility study is completed, and risks are assessed.

3. Design

An architecture is established based on the requirements elaborated earlier. Models, use cases, visual interfaces and test plans are defined.

4. Create

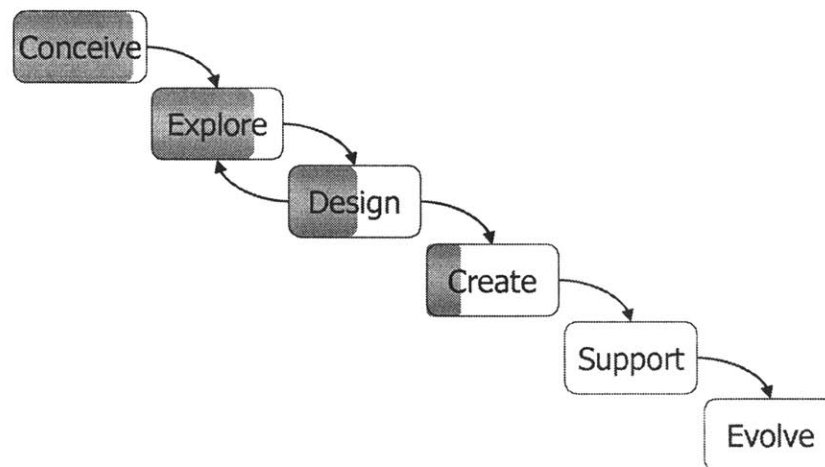
During this phase, the infrastructure is set up, the code and the production graphics are developed, and the database is created. Testing and assessment are also performed.

5. Support

Delivery, implementation, maintenance, hosting/management and enhancement of the system are performed here. Training of the users is also completed.

6. Evolve

This phase proposes a direction for the next potential phases of the solution. The project (solution and requirements) is reviewed. Feedback and lessons learned are collected through surveys and meetings.



* IIS ⇒ Integrated Information Systems, Burlington MA

Figure 7.1: Prototype level of completion (see Appendix I)

The level of completion estimate from IIS is illustrated in Figure 7.1. The percentage of the boxes filled is representative of the estimation. The *Conceive* phase was estimated to be completed at 90%. The *Explore* phase estimate is 75%. The *Design* phase 50% and the *Create* phase 25%. These estimates were based on the material presented during an interview [52].

7.3 Designer Options

Following the consultations with internal (A/A system administrator) and external (Microsoft, Wertheim Inc., and Integrated Informations Systems) experts, two main options were considered for the design of the On-Line TA:

1. Create the On-Line TA at MIT

This option involves the use of MIT students and experts, Microsoft resources and software, and possibly external consultants to add another perspective to the project. MIT students and experts involved are a systems engineering student, a programming student, a part-time System Administrator and a part-time Director of Assessment. From Microsoft, two consultants interacting with MIT are available. The software to be used to create the On-Line TA are Site Server, Source Safe, Front Page and Visual Studio (all Microsoft software). This suite was used to make the Front End for User interface and the Front End authoring tool prototype (see Farnworth [17]). The consultant company is Integrated Informations Systems.

2. Create the On-Line TA externally

This option involves outsourcing the contract to an external company so that minimal supervision has to be maintained by MIT. The company creating the On-Line TA could either work around existing systems like PCDocs Fulcrum or Hummingbird EIP, or use software similar to that used in the previous option. Having the

On-Line TA created by experts minimizes the risk of not having a fully functional system at the right time and minimizes the time that MIT personnel have to spend on the implementation of the On-Line TA (the MIT representative would act as a customer). Cost estimates are provided in the following section.

7.4 Time and Cost Estimates for Implementing the On-Line TA

This section provides time and cost estimates for both of the options previously stipulated (in-house or outsourcing). Most of these estimates are derived from discussions with experts at MIT and in industry.

7.4.1 Estimates for the Option of Creating the On-Line TA at MIT

A time estimation to code and implement the On-Line TA at MIT was established considering the people mentioned in section 7.3. An interview with the A/A System Administrator made it possible to gather the following information [54]:

Table 7.1: Time estimates to create On-Line TA elements at MIT

On-Line TA subsystem	Function	Time estimation (2 people full-time)
Front End	1.0	3-4 days
	4.0	1 week
	11.0	1 week
	12.0	not available
Authoring Tool	3.0	more than one month
	9.0	more than one month
Work Manager	7.0	2 months at most
Project Navigator	7.10	1 month

It was judged necessary to reconsider these estimates because students may face more difficulties than experienced programmers. So the previous estimates were doubled, and

an interval of +/- 1 month was added to each main function. Three more phases were judged necessary to be added:

- Startup (1 month): recruiting programmers, forming the implementation team
- Design Iteration (1 month): improve weaknesses identified during FDR
- Front End Population (2 months): insertion of theory and design processes

The final time estimation is displayed in Figure 7.2.

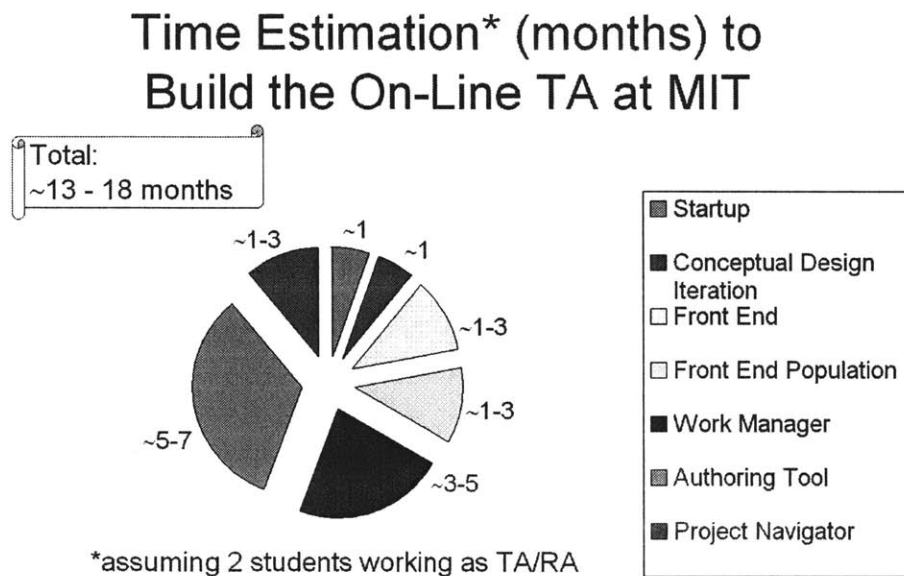


Figure 7.2: Time estimation for the implementation of DE-ICE

The cost estimate was based on the time estimation of Figure 7.2. The stipend of a student working as a TA is \$1,530/month, and the tuition is \$26,050/year (rates of year 1999-2000). Figure 7.3 considers two students receiving thirteen to eighteen months of stipends and one year of tuition. This would be by far the main cost of the project. The estimated hardware cost was judged to be \$12,000 (three servers at \$4,000 each). The use of these servers is explained in Farnworth [17]. \$6,000 was planned for a trip for three persons to Microsoft in Seattle to learn how to use Microsoft Site Server. The external consulting

cost estimation is derived from a proposition of IIS to help MIT complete the *Explore* phase of the design process of Figure 7.1 (see Appendix I). It involves the participation of a Solutions Architect, a Creative Designer and a Project Manager. Three weeks were estimated for the completion of the *Explore* phase. This could occur during the *Conceptual Design Iteration* phase of the time chart of Figure 7.2.

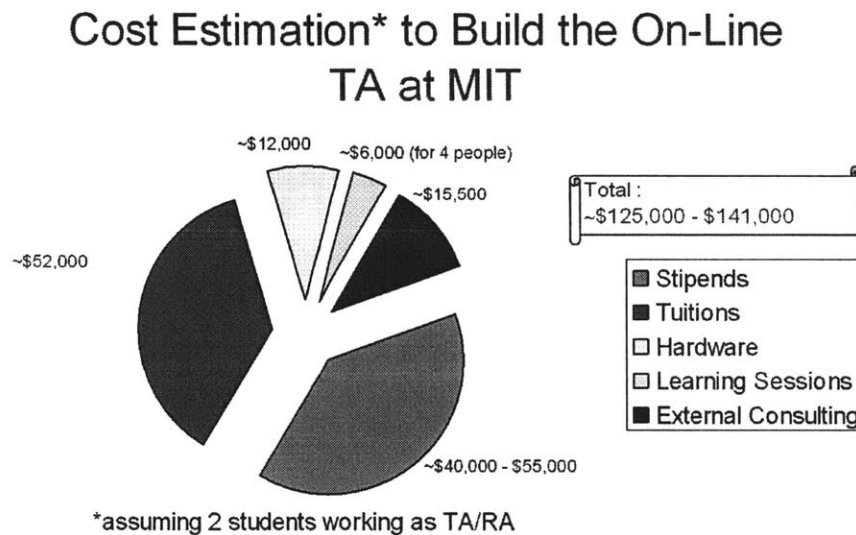


Figure 7.3: Cost estimation for the implementation of DE-ICE

Figure 7.4 illustrates the proposed schedule for implementing the On-Line TA in DE-ICE, assuming that it is created at MIT. The phases of Figure 7.2 are represented by the bars linked together with an arrow in the Gantt chart. The order in which they are accomplished is shown in Figure 7.4. Because of the small number of people at MIT working on the implementation of the On-Line TA and because of the decision to test part of the system during a Fall 2000 class, these phases are not completed concurrently. However, each one is put in the context of the overall system during their implementation to help their inte-

gration.

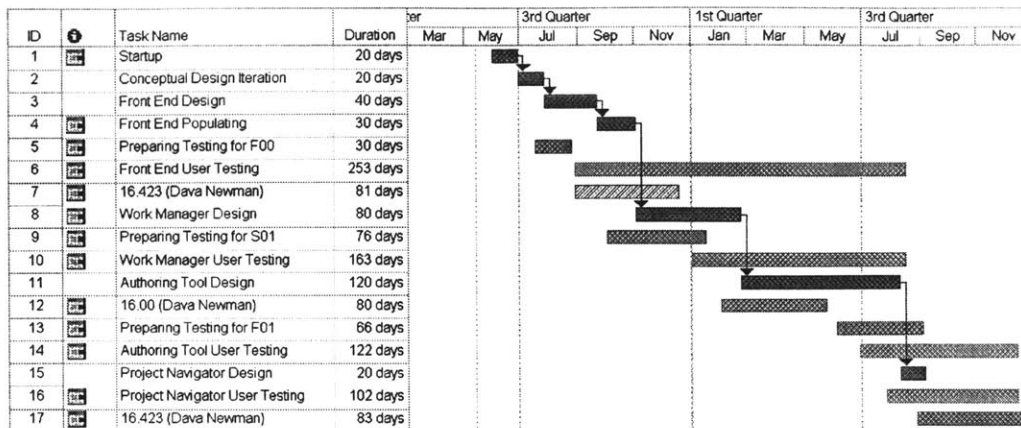


Figure 7.4: Proposed schedule for implementing DE-ICE

Parallel to the main phases are the test preparation and testing phases of each of the main elements of the On-Line TA. Test preparation phases occur in the months prior to the semester in which the tests are conducted. From Figure 7.4, it can be seen that:

- The Front End for User interface will be completed during Summer 2000 and the beginning of Fall 2000. It will be tested during the following year in course 16.423J Aerospace Biomedical and Life Support Engineering, and 16.00 Introduction to Aerospace and Design.
- The Work Manager will be created and implemented during Fall 2000 and the beginning of Winter 2001. It will be tested during Winter 2001 in 16.00 Introduction to Aerospace and Design.
- The third element to be completed will be the Front End Authoring tool, during Winter 2001 and Summer 2001. Its testing will occur during Fall 2001 in the class 16.423J Aerospace Biomedical and Life Support Engineering.
- The project Navigator will be created and implemented at the end of Summer 2001.

It will be tested at the same time as the Front End Authoring tool.

- It is planned that the On-Line TA will be fully operational at the beginning of Winter 2002. It should then be available to other design classes.

7.4.2 Estimates for the Option of Creating the On-Line TA Externally

The other option considered was to ask a private company to create and implement the On-Line TA. In the work proposition returned by IIS, a period of fifteen weeks was judged necessary to complete the *Explore*, *Design* and *Create* phases of the process of Figure 7.1. That means that the system should be ready for the end of August or the beginning of September 2000.

It was anticipated that services for the complete project implementation would be provided by a Solutions Architect, a Consultant, a Designer, a Design Implementer, a Project Manager and a Database Administrator. A final staffing plan would have to be determined at the end of the Explore phase. The estimated total cost was \$321,500 (see Appendix I). It was judged too early for IIS to make any final choice on the software tools to use to create the On-Line TA.

7.4.3 Selected Option

After discussions with MIT staff, it was judged that the cost of doing the On-Line TA externally would be prohibitive. MIT has access to good students and staff members that can accomplish this challenging system design. Collaboration with Microsoft through the I-Campus project for software and human resources positively influenced the decision to create the On-Line TA at MIT.

7.5 The Use, Maintenance and Improvement of the System

A learning curve is associated with the use of the On-Line TA. It is estimated that the students will need a tutorial period of about one and a half hours to learn the main features of

the system.

More time would be required of the System Administrator. It is estimated that he will have to spend about 50% of his time supporting the implementation of the system. Once the On-Line TA is functional, about 20% of his time (five hours) is estimated to be required for routine maintenance.

Time requirements for faculty vary a lot from one professor to another. The time requirements depend on the percentage of the professor's notes already in an electronic form (notes to be inserted in the PDP of the Front End for User interface) and on his level of expertise with the software to be used (mainly web interfaces). The use of teaching assistants is strongly suggested, especially the first time a class uses the On-Line TA. Once a professor gets used to the On-Line TA, it is estimated that the time to interact with the On-Line TA to input new material will not exceed one tenth of the time he normally takes to prepare his class.

As explained in Farnworth [17], DE-ICE captures lessons learned from each project using it. From these lessons learned, faculty may want to improve the PDP they use for their class. A research assistant student could be assigned to test the improvements suggested by the On-Line TA users.

7.6 Learning and Educational Benefits Resulting from DE-ICE

The most important learning and educational benefit from the use of DE-ICE in the A/A department is expected to be a better understanding by the students of the development and the use of design processes to improve the management of a project and the design quality. Having tools, templates and a dedicated facility should make it possible for better project starts and more structured projects throughout their life cycle. Having completed projects in a facility similar to the ones used in aerospace industries, having provided the

students a holistic view of design, and having challenged them in team working situations should better prepare the students to work in the aerospace industry.

7.7 Future Work/Recommendations

This section presents future work to be done and recommendations for both the On-Line TA and DE-ICE.

7.7.1 Future Work To Be Done on the On-Line TA

As mentioned by a faculty member during the final FDR presentation, it seemed that the connection between DE-ICE and the On-Line TA was not absolutely clear, even with the matrix presented in Figure 6.1. A refinement of the needs and the technical requirements specifically for the On-Line TA will have to be done in order to fully demonstrate the usefulness of the On-Line TA to the faculty. This will be achieved during the conceptual design iteration phase of Figure 7.2.

After being officially approved by the faculty, the On-Line TA will have to be fully coded and tested. This will occur over a period of 18 months (see Figure 7.4). The On-Line TA team will need formation on the software and hardware to be used (like Site Server, SQL databases, server configuration). This is judged necessary due to the complexity of Microsoft Site Server and the difficulty of maintaining a stable development environment for the Front End for User interface and the Front End Authoring tool (see Farnworth [17]). Hardware will need to be bought and installed. The overall system will need to be configured and tested. The metrics developed in the QFD matrix (see Figure 4.1) will have to be refined so that quantitative measurements could come out of the tests. These measurements will be used to further improve the system.

7.7.2 Future Work To Be Done on DE-ICE

More feedback from faculty and students will have to be obtained on the final architecture

selected (ABC'), mainly on the approach of including the purchase of a laptop in the student's tuition. This feedback could be obtained through informative meetings or through surveys. A final decision on this topic will have to be made.

The distance collaborative system will have to be selected. Preliminary tests have already been made on PicTel and NetMeeting (see Nolet [31]), but further investigations will have to be completed before finalizing the choice. Inexpensive PC cameras could be used in between to maintain a minimal video connection with external sites.

Further tests of the DOME system will have to be made before integrating it in DE-ICE, since it is still under development. The tests should demonstrate that this environment will not add more constraints or more work to the users.

Once the construction of the Learning Laboratory is completed, elements identified in Architecture ABC' could be installed. This process could occur over an extended period while the laboratory is being monitored and feedback from the users is received. The laboratory could eventually look like the DE-ICE Architecture ABC'.

7.8 Conclusion

Among the reforms adopted by the A/A department to converge toward the CDIO initiative was the intention of the A/A department to conceive DE-ICE. DE-ICE will be used by students and faculty in future design classes. A group of MEng students formed the DE-ICE team and took up this challenging design project. They identified the following mission statement: *To develop an operational framework for a design center to enhance learning in an academic environment.*

Two main goals were identified. The first one was to develop recommendations for the architecture of the design environment. To do so, a systems engineering process was followed. Benchmarking of existing systems was completed, like JPL's PDC, Aerospace

Corporation's CDC, TRW's ICDF, Caltech's LSMD, and more. Educational needs for DE-ICE included:

1. Active learning in a design environment
2. Holistic view of design
3. Improved knowledge of and experience with the design process
4. Support of life cycle analysis
5. Improved quality of student design work
6. Increased productivity for given amount of time
7. Support for team enablers
8. Sustainable system
9. Highly usable system

The support of MIT's operational modes was identified as a constraint and influenced the choice of the final technical requirements.

Using a QFD matrix, technical requirements addressing the previous needs were identified, classified and selected. The ones retained were:

1. Design and analysis support
2. Provide guidance throughout the design process
3. Planning and management of the design process
4. Experimentation support
5. Ability to operate on any platform
6. Distance collaborative support
7. Flexible system
8. Presentation and reporting support

Conflicts between these requirements were mentioned, and some preliminary metrics were elaborated. A use case study was prepared in order to characterize the way the users directly interact with the system.

Architectural elements were mapped with the main technical requirements in a product matrix. From this assessment, four architectures were elaborated: three preliminary ones and a final one (Architecture ABC') that combined elements of the three preliminary architectures. These architectures were evaluated and compared using Pugh's matrix to the Baseline Architecture initially planned by the A/A department. Architecture ABC' was the one recommended for DE-ICE architecture. The evaluation showed major improvements from the Baseline Architecture.

The second goal of the project was to discover a key enabler of the system and to develop a component. Different prototype ideas were defined. A matrix was developed to assess each prototype idea with the technical requirements retained. The On-Line TA was finally selected as being the DE-ICE key enabler to be made into a prototype.

Prototype elements were identified, and their design was split among team members. The author elaborated the Work Manager that enables students to manage their work, whether they are working in teams or by themselves, on campus, from home or even through the Internet, using any kind of platforms (PC, Unix or Macintosh). Functionality of the Work Manager was developed, and a demonstration was made during the team's final presentation.

The feedback received by the team was very positive throughout the semester. Professor Joel Sercel from Caltech even stated that "The On-Line TA could revolutionize engineering education as we know it."

The author further elaborated on an implementation plan of the On-Line TA. Options of creating the On-Line TA internally or externally using a consultant company were

described. Time and cost estimates for each option, as well as implementation schedules, were developed. The option of creating the On-Line TA at MIT was finally selected. Future work to fully implement the system was mentioned.

This project was a very important step toward integrated concurrent engineering teaching in the MIT A/A department. It has paved the way for the creation of a design center to be used in academia. Once DE-ICE is implemented, it will allow students to do better projects and better quality designs and to gain an expertise that was previously only available in industry and government.

References

- [1] P. C. Wankat, F. S. Oreovicz. *Teaching Engineering*, McGraw-Hill Inc., New York, 1993, 370 pages.
- [2] D. S. Goldin. Tools of the Future, *Journal of Engineering Education*, January 1999, pp. 31-35.
- [3] A. H. Slocum. *Precision Machine Design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992, pp. 17-31.
- [4] C. H. Amon, S. Finger, D. P. Siewiorek, A. Smailagic. Integrating Design Education, Research and Practice at Carnegie Mellon: A Multi-Disciplinary Course in Wearable Computers, *Journal of Engineering Education*, October 1996, pp. 279-285.
- [5] L. G. Richards, M. Gorman, W. T. Scherer, R. D. Landel. Promoting Active Learning with Cases and Instructional Modules, *Journal of Engineering Education*, October 1995, pp. 375-381.
- [6] D. N. Mavris, D. A. DeLaurentis, M. A. Hale, J. C.M. Tai. *Elements of an Emerging Virtual Stochastic Life Cycle Design Environment*, 1999-01-5638, 1999 World Aviation Conference, San Francisco, California, October 19-21, 1999, 14 pages.
- [7] IMAGE web site, <http://alpha.cad.gatech.edu/image/index.html>, Aerospace Systems Design Laboratory, Georgia Institute of Technology, August 2000.
- [8] F. Pahng, N. Senin, D. Wallace. *Distributed Object-Based Modeling and Evaluation of Design Problems*, <http://cadlab.mit.edu/publications/98-pahng-dome/abstract.html>, MIT's Computer-Aided Design Laboratory, August 1998.
- [9] J. Neff, S. P. Presley. *Implementing a Collaborative Conceptual Design System*, f134-spp.doc©2000 IEEE, The Aerospace Corporation, El Segundo, California.
- [10] C. Williamson, J. T. Bernhard, K. Chamberlin. Perspectives on an Internet-Based Synchronous Distance Learning Experience, *Journal of Engineering Education*, January 2000, pp. 53-61.
- [11] M. A. Hale, D. N. Mavris, D. L. Carter. *The Implementation of a Conceptual Aerospace Systems Design and Analysis Toolkit*, SAE/AIAA 1999-01-5639, 1999, 7 pages.
- [12] J. C. Heim, K. K. Parsons, S. F. Sepahban, R. C. Evans. *TRW Process Improvements for Rapid Concept Designs*, TRW Space & Technology Division, Redondo Beach, California, 9 pages.
- [13] *Concept Design Center (CDC) User's Guide*, prepared by the CDC Engineering and Technology Group, Aerospace Report No. ATR-98 (8227)-1, The Aerospace Corporation, El Segundo, California, August 25, 1998, 25 pages.
- [14] J. L. Smith. *Concurrent Engineering in the Jet Propulsion Laboratory Project Design Center*, Society of Automotive Engineers Inc., 98AMTC-83, 1998, 14 pages.

- [15] *Project Design Center*, <http://pdc.jpl.nasa.gov/>, July 2000.
- [16] S. Wakayama. *Multidisciplinary Design Optimization of the Blended-Wing-Body*, AIAA-98-4938, 9 pages.
- [17] B. J. Farnworth. *Architectural Framework to Support Integrated Concurrent Engineering in an Academic Institution*, Master of Engineering thesis, Massachusetts Institute of Technology, June 2000.
- [18] LSMD web site, <http://www.lsmd.caltech.edu/>, California Institute of Technology, August 2000.
- [19] *NASA Systems Engineering Handbook*, SP-6105, PPMI, National Aeronautics and Space Administration, June 1995, 154 pages.
- [20] K. T. Ulrich, S. D. Eppinger. *Product Design and Development*, McGraw-Hill Inc., New York, 1995, 289 pages.
- [21] S. Pugh. *Total Design, Integrated Methods for Successful Product Engineering*, Addison-Wesley Publishers Ltd., Wokingham, England, 1991, 278 pages.
- [22] A. K. Manka. *Developing a Design Environment for Integrated Concurrent Engineering (DE-ICE) in University Education: Integrating Student Designers, Design Tools, and Active Learning*, Master of Engineering thesis, Massachusetts Institute of Technology, June 2000.
- [23] M. W. Goldberg, S. Salari. *An Update on WebCT (World-Wide-Web Course Tools) - a Tool for the Creation of Sophisticated Web-Based Learning Environments*, Proceedings of NAUWeb '97 - Current Practices in Web-Based Course Development, Flagstaff, Arizona, June 12-15, 1997.

Additional Sources of Information

- [24] *Educating Engineers for the Real World, a proposal by the Department of Aeronautics and Astronautics to Build the Teaching Laboratory for Complex Systems*, Submitted to the Committee for Review of Space Planning, October 9, 1998.
- [25] E. F. Crawley. Presentation made at the faculty workshop held at the University Park Hotel @ MIT, May 19, 2000.
- [26] *Microsoft Proposal*, Submitted by the Department of Aeronautics and Astronautics, October 4, 1999.
- [27] E. F. Crawley. *CDIO Syllabus for SOE*, May 15, 2000.
- [28] D. Frey, I. A. Waitz, R. J. Hansman, C. R. A. Hallam, B. Linder. *Outcome Assessment for the CDIO Initiative*.
- [29] E. M. Murman, R. J. Hansman, J. P. Clarke. *Aircraft System and Product Development: Teaching the Conceptual Phase*, Abstract Submitted to AIAA 39th Aerospace Sciences Meeting Reno. NV, January 8-11, 2001.
- [30] B. Linder. *MIT Faculty Newsletter*, Promoting Active Learning, January/February 2000, p. 9.
- [31] S. Nolet, D. Brodeur, D. M. Qualters, D. J. Newman. *Microsoft I-Campus: Phase I*, Submitted by the Department of Aeronautics and Astronautics, June 16, 2000, 58 pages.
- [32] L. Brooks, H. Webb. *Information Technology Support for the Learning Organisation*, Department of Computer Science, University of York.
- [33] R. C. Larson. A Week in the Life of CAES (Center for Advanced Educational Services), *MIT Faculty Newsletter*, Vol. XII, no. 2, November/December 1999, pp. 28-29.
- [34] *Integrated Concept Development Facility (ICDF), Capabilities and Resources*, presentation made at TRW for the Quality Day on October 20, 1999.
- [35] *An Introduction to The Concept Design Center*, video produced by The Aerospace Corporation, El Segundo, California, trt: 10:50.
- [36] J. A. Aguilar, A. B. Dawdy, G. W. Law. *The Aerospace Corporation's Concept Design Center*, The Aerospace Corporation, El Segundo, California.
- [37] J. L. Smith. *Project Design Center, Overview*, presentation made at JPL, January 9, 1999.
- [38] B. Oberto. *Team-X/A/T, Advanced Conceptual Design and Technology*, design team workshop presentation, Jet Propulsion Laboratory, December 9, 1999.
- [39] J. C. Sercel. *Introduction to Integrated Concurrent Engineering (ICE), ICEMaker*,

- and Drawcraft*, presentation made in the class 16.89 Space Systems Engineering, Massachusetts Institute of Technology, March 29, 2000.
- [40] J. Sercel, S. D. Wall. *Integrated Concurrent Engineering (ICE) Design Centers*, California Institute of Technology, 1998.
 - [41] B. Fredriksson. *Holistic systems engineering in product development*, SAAB Military Aircraft, Sweden, pp. 23-31.
 - [42] D. J. Newman. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 9, 2000.
 - [43] D. W. Miller. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 7, 2000.
 - [44] R. J. Hansman. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 2000.
 - [45] M. Drela. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 2000.
 - [46] D. E. Hastings. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 2000.
 - [47] P. W. Young. Interview performed by Simon Nolet, Massachusetts Institute of Technology, March 2000.
 - [48] K. Blair. Interview performed by Bruce Farnworth, Alex Manka and Simon Nolet, Massachusetts Institute of Technology, April 6, 2000.
 - [49] *Project MINERVA: Establish an Enabling Space Infrastructure that will Support the Exploration of Mars*, 16.89 Space Systems Engineering, design report presented to Professors D. Hastings, J. Warmkessel, L. Fesq, D. Miller and Colonel J. Keesee, Spring 2000.
 - [50] C. W. Boppe. *16.870 Aerospace Product Design*, class notes, September 1999.
 - [51] C. R. A. Hallam. *Astronet, The Computing Environment for Aero/Astro at MIT*, presentation made to MIT IS and the Learning Lab Design Team, Summer 1999.
 - [52] K. Rankin, Enterprise Solutions Executive. Interview performed by Simon Nolet, Integrated Information Systems, Burlington, Massachusetts, May 12, 2000.
 - [53] K. Rankin, Enterprise Solutions Executive. *MIT-Aerospace OLTA (On-Line Teaching Assistant)*, work proposition, Integrated Informations Systems, Burlington, Massachusetts, May 15, 2000, 15 pages.
 - [54] F. J. Donovan, System Administrator. Interview performed by Simon Nolet, Massachusetts Institute of Technology, May 12, 2000.

Appendix A

CDIO Syllabus

E. F. Crawley. *The CDIO Syllabus: A Statement of Goals for Engineering Undergraduate Education*, MIT CDIO Report #1, Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, June 2000.

In contemporary engineering undergraduate education, there is a seemingly irreconcilable tension between two growing needs. On one hand, there is the ever increasing body of technical knowledge that it is felt that a graduating student must command. On the other hand, there is a growing recognition that a young engineer must possess a wider array of personal, interpersonal and system building knowledge and skills that will allow them to function within real engineering teams and processes, producing real products and systems. In order to reconcile these disparate needs, we must develop: a new codified understanding of the skills needed by the contemporary engineer; new approaches to enabling and enhancing the learning of these skills; and new systems to assess learning and improve our education. These are the broad goals of the CDIO program at MIT (the origin of this name will become obvious shortly). The first tangible outcome of this program is the CDIO Syllabus.

The primary objective of the CDIO Syllabus is to formally summarize a set of knowledge, skills and attributes that alumni, industry and academia desire in young engineers. The Syllabus can be used to prioritize the personal, interpersonal and system building skills among themselves, and relative to the "traditional" body of technical knowledge and reasoning. Further, the syllabus can be used to create and define new educational initiatives, and as a basis for a rigorous assessment process.

E. F. Crawley. *CDIO Syllabus*, V3.2.1, May 25, 2000.

1 TECHNICAL KNOWLEDGE AND REASONING

1.1 KNOWLEDGE OF UNDERLYING SCIENCES [a]

1.1.1 Mathematics (including statistics)

1.1.2 Physics

- 1.1.3 Chemistry
- 1.1.4 Biology
- 1.2 CORE ENGINEERING FUNDAMENTAL KNOWLEDGE [a]
 - 1.2.1 Fluid Mechanics
 - 1.2.2 Solid Mechanics and Materials
 - 1.2.3 Dynamics
 - 1.2.4 Signals and Systems
 - 1.2.5 Thermodynamics
 - 1.2.6 Control
 - 1.2.7 Computers and computation
- 1.3 PROFESSIONAL ENGINEERING KNOWLEDGE AND SKILLS [k]
 - 1.3.1 Aerodynamics
 - 1.3.2 Structural Mechanics
 - 1.3.3 Structures and Materials
 - 1.3.4 Jet and Rocket Propulsion
 - 1.3.5 Flight and Advanced Aerospace Dynamics
 - 1.3.6 Computational Techniques
 - 1.3.7 Estimation and Navigation
 - 1.3.8 Human and Supervisory Control
 - 1.3.9 Digital Communications
 - 1.3.10 Software Engineering
 - 1.3.11 Autonomy
 - 1.3.12 Digital Circuits and Systems

2 PERSONAL AND PROFESSIONAL SKILLS AND ATTRIBUTES

- 2.1 ENGINEERING REASONING AND PROBLEM SOLVING (4.0) [e]
 - 2.1.1 Problem Identification and Framing (4.4/4)
 - Evaluate data and symptoms
 - Analyze assumptions and bias
 - Examine issue prioritization in context of overall goals
 - Formulate a plan of attack (incorporating model, analytical and numerical solutions, algorithms and qualitative analysis, experimentation and consideration of uncertainty)
 - 2.1.2 Modeling (4.3/4)
 - Employ appropriate assumptions to simplify complex systems and

- environment
 - Choose and apply conceptual and qualitative models
 - Choose and apply quantitative models and simulations
- 2.1.3 Estimation and Qualitative Analysis (4.0/4)
 - Estimate the order of magnitude, bounds and trends
 - Analyze possible errors (limits, units, etc.)
 - Demonstrate the generalization of appropriate analytic solutions
- 2.1.4 Analysis With Uncertainty (3.7/4)
 - Inventory incomplete and ambiguous information
 - Apply probability and statistics of events and sequences
 - Practice engineering cost-benefit or risk analysis
 - Discuss decision analysis
 - Schedule margins and reserves
- 2.1.5 Closing the Problem (3.8/4)
 - Analyze essential results of analysis and test data
 - Analyze and reconcile discrepancies in results
 - Synthesize solutions
 - Formulate recommendations
 - Appraise the problem solving process used
- 2.2 EXPERIMENTATION AND KNOWLEDGE DISCOVERY (3.3) [b]
 - 2.2.1 Hypothesis Formulation (3.4/3)
 - Select the critical questions to be examined
 - Formulate hypotheses to be tested
 - Explain the need for controls
 - 2.2.2 Survey of Print and Electronic Literature (3.0/3)
 - Choose the literature research strategy
 - Identify information using library tools (on-line catalogs, electronic databases, search engines)
 - Locate, sort and classify the primary information
 - Question the quality and reliability of information
 - Discriminate the essentials and innovations contained in the information
 - Identify research questions that are unanswered
 - Record citations to references
 - 2.2.3 Experimental Inquiry (3.6/4)
 - Design the experimental concept and strategy
 - Discuss the precautions when humans are used in experiments
 - Execute experiment construction
 - Execute test protocols and experimental procedures

- Execute and record experimental measurements
- Analyze and report experimental data
- Compare experimental data vs. available models

2.2.4 Hypothesis Test, and Defense (3.3/3)

- Discuss the statistical validity of data
- Discuss the limitations of data employed
- Prepare conclusions, supported by data, needs and values
- Appraise knowledge discovery process used

2.3 SYSTEM THINKING (2.8)

2.3.1 Thinking Holistically (2.9/3)

- Identify and define a system, its behavior, and its elements
- Use trans-disciplinary approaches that insure the system is understood from all relevant perspectives
- Identify the societal, enterprise and technical context
- Identify the interactions external to the system, and the behavioral impact of the system

2.3.2 Emergence and Interactions in Systems (2.6/3)

- Describe the abstractions necessary to define and model system
- Identify the importance of the interfaces among elements
- Identify the behavioral and functional (intended and unintended) properties which emerge from the system
- Recognize evolutionary adaptation over time

2.3.3 Prioritization and Focus (2.7/3)

- Indicate all factors relevant to the system in the whole
- Identify the driving factors from among the whole
- Explain energy and resource allocations to resolve the driving issues

2.3.4 Trade-offs and Balance (3.1/3)

- Identify tensions and factors to trade-off
- Choose and employ solutions that balance various factors, resolve tensions and optimize the system as a whole
- Describe flexible vs. optimal solutions over the system lifetime
- Appraise the system thinking used

2.4 PERSONAL SKILLS AND ATTRIBUTES (3.4)

2.4.1 Initiative and Willingness to Take Risks (3.4/3)

- Identify needs and opportunities for risks and initiative
- Explain potential benefits and risks of an action
- Explain the methods and timing of project initiation
- Practice leadership in new endeavors, with a bias for appropriate action

Practice definitive action, delivery of results and reporting on actions

2.4.2 Perseverance and Flexibility (3.8/4)

Demonstrate self-confidence, enthusiasm, and passion

Demonstrate the importance of hard work, intensity and attention to detail

Identify issues vital for success

Demonstrate adaptation to change

Demonstrate a willingness to work with others, consider and embrace various viewpoints

Demonstrate an acceptance of criticism and positive response

Discuss the balance between personal and professional life

2.4.3 Creative Thinking (3.6/4)

Demonstrate conceptualization and abstraction

Demonstrate synthesis and generalization

Use the process of invention

2.4.4 Critical Thinking (3.8/4)

Analyze the statement of the problem

Choose logical arguments and solutions

Evaluate evidence

Locate contradictory perspectives, theories and facts

Identify logical fallacies

Test hypotheses and conclusions

2.4.5 Personal Inventory (2.9/3)

Describe one's skills, interests, strengths, weaknesses

Explain one's skills, interests and strengths

Discuss the extent of one's abilities, and accept responsibility for self-improvement to overcome important weaknesses

Discuss the importance of both a depth and breadth of knowledge

2.4.6 Curiosity and Lifelong Learning (3.1/3) [i]

Discuss the motivation for continued self-education

Practice working independently

Demonstrate the skills of self-education

Express an awareness of one's own learning style

Explain developing mentors

2.4.7 Time and Resource Management (3.3/3)

Discuss task prioritization

Explain important and/or urgent tasks

Explain efficient execution

2.5 PROFESSIONAL SKILLS AND ATTITUDES (3.0)

- 2.5.1 Professional Ethics, Integrity, Responsibility and Accountability (3.7/4) [f]
 - Analyze one's ethical standards and principles
 - Demonstrate the courage to act on principle despite adversity
 - Recognize the possibility of conflict between professionally ethical imperatives
 - Demonstrate an understanding that it is acceptable to make mistakes, but accept accountability for them
 - Practice proper allocation of credit and collaboration
 - Demonstrate a commitment of service
- 2.5.2 Professional Behavior (2.7/3)
 - Describe a professional bearing
 - Explain professional courtesy
 - Identify international customs and norms of interpersonal contact
- 2.5.3 Proactively Planning for One's Career (2.7/3)
 - Describe a personal vision for future
 - Describe network with professionals
 - Define one's portfolio of professional skills
- 2.5.4 Stay Current on World of Engineer (2.9/3)
 - Discuss the potential impact of new scientific discovery
 - Describe the social and technical impact of new technologies and innovations (e.g. IT)
 - Discuss a familiarity with current practices/technology in engineering
 - Explain the links between engineering theory and practice

3 INTERPERSONAL SKILLS: TEAMWORK AND COMMUNICATION

- 3.1 MULTI-DISCIPLINARY TEAMWORK(3.3) [d]
 - 3.1.1 Form Effective Teams (3.4/3)
 - Identify the stages of team formation and life cycle
 - Interpret task and team processes
 - Identify team roles and responsibilities
 - Inventory and analyze the individual goals, needs and characteristics (works styles, cultural differences) of the team
 - Inventory and analyze the strengths and weakness of the team
 - Discuss ground rules on norms of team confidentiality, accountability and initiative
 - 3.1.2 Team Operation (4.0/4)
 - Choose goals and agenda
 - Execute the planning and facilitation of effective meetings
 - Apply team ground rules

- Practice effective communication (active listening, collaboration, providing and obtaining information)
- Demonstrate positive and effective feedback
- Practice the planning, scheduling and execution of a project
- Create solutions to problems (team creativity and decision making)
- Practice conflict negotiation and resolution
- 3.1.3 Team Growth and Evolution (2.7/3)
 - Describe strategies for reflection, assessment, and self-assessment
 - Identify skills for team maintenance and growth
 - Identify skills for individual growth within the team
 - Explain strategies for team communication and writing
- 3.1.4 Leadership (3.4/3)
 - Explain goals and objectives
 - Practice team process management
 - Practice leadership and facilitation styles (directing, coaching, supporting, delegating)
 - Explain approaches to motivation (incentives, example, recognition, etc)
 - Practice team representation to the larger enterprise
 - Describe mentoring and counseling
- 3.1.5 Technical Teaming (3.0/3)
 - Describe working in different types of teams:
 - Cross-disciplinary (include non-engineer) teams
 - Small team vs. large team
 - Distance, distributed and electronic environments
 - Demonstrate technical collaboration with team members
- 3.2 COMMUNICATIONS (3.6) [g]
 - 3.2.1 Communications Strategy (3.5/4)
 - Analyze a communication situation
 - Choose communications objectives
 - Analyze the audience
 - Analyze context
 - Choose a communications strategy
 - Choose media
 - Choose a style (proposing, reviewing, collaborating, documenting, teaching)
 - Select content and organization
 - 3.2.2 Communications Structure (3.8/4)
 - Construct logical, persuasive arguments
 - Create the appropriate structure and relationship amongst ideas
 - Choose relevant, credible, accurate support evidence

Practice conciseness, crispness, precision and clarity of language

Analyze rhetorical factors (e.g. audience bias)

Recognize cross-disciplinary cross-cultural communications

3.2.3 Written Communication (3.9/4)

Demonstrate writing with coherence and flow

Demonstrate formatting

Practice writing without spelling, punctuation or grammatical errors

Demonstrate technical writing

Apply various styles (informal, formal memos, reports, etc)

3.2.4 Electronic/Multimedia Communication (3.1/3)

Identify the norms associated with the use of E-mail, voice mail, and videoconferencing

Discuss writing supporting technical documentation

Demonstrate electronic and web presentations

3.2.5 Graphical Communication (3.4/3)

Demonstrate sketching and drawing

Interpret formal drawing, drafting and rendering

Demonstrate construction of tables, graphs and charts

3.2.6 Oral Presentation and Inter-Personal Communications (4.1/4)

Prepare presentation and support media

Use with appropriate language, style, timing and flow

Use appropriate nonverbal communications (gestures, eye contact, poise)

Demonstrate answering questions

4 CONCEIVING, DESIGNING, IMPLEMENTING AND OPERATING SYSTEMS IN THE ENTERPRISE AND SOCIETAL CONTEXT

4.1 EXTERNAL AND SOCIETAL CONTEXT (2.0) [h]

4.1.1 Roles and Responsibility of Engineers (2.2/2)

Define the goals and roles of the engineering profession

Describe the responsibilities of engineers to society

4.1.2 Understand the Impact of Engineering (2.5/3)

Explain the impact of engineering on the environment, society, and modern culture (including financial and economic factors)

4.1.3 Understand How Engineering Is Regulated (1.7/2)

Recognize the way in which legal and political systems regulate and influence engineering

Describe how professional societies license and set standards

Describe intellectual property

- 4.1.4 Knowledge of Historical and Cultural Context (1.4/2)
 - Describe the diverse nature and history of human societies as well as their literary, philosophical, and artistic traditions
 - Recognize use of discourse and analysis appropriate to the discussion of language, thought and values
- 4.1.5 Knowledge of Contemporary Issues and Values (2.2/2) [j]
 - Describe the important contemporary political, social, legal and environmental issues and values
 - Recognize the mechanisms for expansion and diffusion of knowledge
- 4.1.6 Developing a Global Perspective (2.1/2)
 - Recognize the internationalization of human activity
 - Recognize international cultures and norms
- 4.2 ENTERPRISE AND BUSINESS CONTEXT (1.8)
 - 4.2.1 Appreciating Different Enterprise Cultures (1.6/2)
 - Recognize the differences in process, culture, and metrics of success in various cultures:
 - Academic vs. corporate vs. governmental vs. non-profit/NGO
 - Market vs. policy driven
 - Large vs. small
 - Centralized vs. distributed
 - Research and development vs. operations
 - Mature vs. growth phase vs. entrepreneurial
 - Longer vs. faster development cycles
 - 4.2.2 Enterprise Strategy, Goals, and Planning (2.2/2)
 - State the mission and scope of the business
 - Recognize an enterprise's core competence and markets
 - Recognize the research and technology process
 - Recognize key alliances and supplier relations
 - List financial and managerial goals and metrics
 - Recognize financial planning and control
 - List the stake-holders (owners, employees, customers, etc)
 - 4.2.3 Technical entrepreneurship (1.8/2)
 - Recognize entrepreneurial opportunities that can be addressed by technology
 - Recognize technologies that can create new products
 - Recognize the issues in entrepreneurial finance and organizations
 - 4.2.4 Working successfully in Organizations (1.8/2)
 - Define management function
 - Describe various roles and responsibilities in an organization
 - Describe the roles of functional and program organizations

Recognize how to work within hierarchy and organizations
Recognize change, dynamics and evolution in organizations

4.3 CONCEIVING AND ENGINEERING SYSTEMS (3.1) [c]

4.3.1 Setting System Goals and Requirements (3.2/3)

Identify market needs and opportunities

Elicit and interpret customer needs

Identify opportunities which derive from new technology or latent needs

Explain contextual factors

Identify enterprise goals, strategies, capabilities and alliances

Locate competitors and benchmarking information

Discuss ethical, social, environmental, legal and regulatory issues

Explain the probability of change in the factors which influence the system, its goals and resources available

Express system goals and requirements

Indicate the language/format of goals and requirements

Interpret initial goals

Describe system performance metrics

Interpret requirements for completeness and consistency

4.3.2 Defining Function, Concept and Architecture (3.2/3)

Identify necessary system functions (and behavioral specifications)

Analyze system concepts

Identify the appropriate level of technology

Analyze trade-offs among and recombination of concepts

Identify high level architectural form and structure

Discuss the decomposition of form into elements, assignment of function to elements (HW and SW), & definition of interfaces

4.3.3 Modeling of System and Insuring Goals Can Be Met (3.1/3)

Locate appropriate models of technical performance

Describe concept of implementation and operations

Describe life cycle value and costs (design, implementation, operations, opportunity, etc.)

Indicate trade-offs among various goals, function, concept and structure and iterate until convergence

4.3.4 Project Management (3.0/3)

Describe project control for cost, performance, and schedule

Indicate appropriate transition points and conduct reviews

Explain configuration management and documentation

Record performance against baseline

Define earned value recognition

Discuss the estimation and allocation of resources

Identify risks and alternatives

Recognize development process improvement

4.4 DESIGNING (3.4) [c]

4.4.1 The Design Process (3.9/4)

Translate system level goals and requirements into requirements for each element or component

Examine alternatives

Analyze alternatives

Resolve trade-offs

Experiment with prototypes and test articles

Apply appropriate optimization in the presence of constraints

Demonstrate iteration until convergence

Synthesize final design

Demonstrate accommodation of rapid/major changing requirements

4.4.2 The Design Process Phasing and Approaches (2.9/3)

Explain the activities in the phases of system design (e.g. conceptual, preliminary, and detailed design)

Describe process models appropriate for particular development projects (waterfall, spiral, concurrent, etc.)

Describe the process for single, platform and derivative products

4.4.3 Utilization of Knowledge in Design (3.4/3)

Exploit technical and scientific knowledge

Employ creative and critical thinking, and problem solving

Discuss prior work in the field, standardization and reuse of designs (including reverse engineer)

Discuss design knowledge capture

4.4.4 Disciplinary Design (3.4/3)

Choose appropriate techniques, tools, and processes

Describe tool calibration and validation

Analyze alternatives quantitatively

Practice modeling, simulation and test

Describe analytical refinement of the design

4.4.5 Multidisciplinary Design (3.4/3)

Identify interactions between disciplines

Indicate dissimilar conventions and assumptions

Explain differences in the maturity of models

Describe multidisciplinary design environments

4.4.6 Multi-Objective Design (DFX) (3.5/4)

Illustrate design for:

Performance, life cycle cost and value

Aesthetics and human factors
Implementation, verification, test and environmental sustainability
Operations
Maintainability, reliability, and safety
Robustness, evolution, product improvement and retirement

4.5 IMPLEMENTING (2.3) [c]

- 4.5.1 Designing and Modeling of the Implementation Process (2.3/2)
 - State the goals metrics for implementation performance, cost and quality
 - Recognize the implementation system design:
 - Task and cell/unit layout
 - Work flow
 - Considerations for human user/operators
- 4.5.2 Hardware Manufacturing Process (2.1/2)
 - Describe the manufacturing of parts
 - Describe the assembly of parts into larger constructs
 - Define tolerances, variability, key characteristics and statistical process control
- 4.5.3 Software Implementing Process (2.4/2)
 - Explain the break down of high level components into module designs (including algorithms and data structures)
 - Discuss algorithms (data structures, control flow, data flow)
 - Describe the programming language
 - Code the low-level design
 - Describe the system build
- 4.5.4 Hardware Software Integration (2.4/2)
 - List issues of integration of SW in electronic hardware (size of processor, communications, etc)
 - List issues of SW integration with sensor, actuators and mechanical HW
 - Describe HW/SW function and safety
- 4.5.5 Test, Verification, Validation, and Certification (2.7/3)
 - Explain verification methods selection
 - Discuss test and analysis procedures (hardware vs. software, acceptance vs. qualification)
 - Report on the traceability and verification of requirements
 - Explain the certification to standards
- 4.5.6 Managing Implementation (2.0/2)
 - Describe the organization and structure for implementation
 - Discuss sourcing, partnering, and supply chain

- Recognize control of implementation cost, performance and schedule
- Describe quality and safety assurance
- Describe implementation process improvements

4.6 OPERATING (2.2) [c]

4.6.1 Modeling, Designing and Optimizing Operations (2.6/3)

- Interpret the goals for operational performance, cost, and value, and identify metrics

- Explain operations process architecture and development

- Explain operations (and mission) analysis and modeling

4.6.2 Training and Operations (2.2/2)

- Describe training for professional operations:

 - Simulation

 - Instruction and programs

 - Procedures

- Recognize education for consumer operation

- Describe execution of operations processes

- Recognize operations process interactions

4.6.3 Supporting the System Lifecycle (2.4/2)

- Explain maintenance and logistics

- Describe assessment of performance and reliability

- Describe assessment of life cycle value and costs

- Explain feedback to system improvement

4.6.4 System Improvement and Evolution (2.4/2)

- Define pre-planned product improvement

- Recognize improvements based on needs observed in operation

- Recognize system upgrades

- Recognize operational work-arounds

4.6.5 Disposal and Life-End Issues (1.5/2)

- Define the end of useful life

- List disposal option

- Define residual value

- List environmental considerations

4.6.6 Operations Management (2.3/2)

- Describe the organization and structure for operations

- Recognize partnering and alliances

- Recognize control of operations cost, performance and scheduling

- Describe quality and safety assurance

- Recognize operations process improvement

- Define life cycle management

Appendix B

Design Engineering References

B.1 Reference Books Suggested by Slocum [3]

1. T. Busch, *Fundamentals of Dimensional Metrology*, Delmar Publishers.
2. *Machine Tool Specs*, Huebner Publishing Co.
3. J. Lienhard, *A Heat Transfer Textbook*, Prentice Hall.
9. W. Moore, *Foundations of Mechanical Accuracy*, Moore Special Tool Co.
10. E. Oberg et al., *Machinery's Handbook*, Industrial Press.
11. E. P. Popov, *Engineering Mechanics of Solids*, Prentice Hall.
12. J. Shigley and C. Mischke, *Standard Handbook of Machine Design*, McGraw-Hill Book Co.
13. M. F. Spotts, *Design of Machine Elements*, Prentice Hall.
14. R. Steidel, Jr., *An Introduction to Mechanical Vibrations*, John Wiley & Sons
15. M. Weck, *Handbook of Machine Tools*, John Wiley & Sons.
16. W. Woodson, *Human Factors Design Handbook*, McGraw-Hill Book Co.
17. *The Principles and Techniques of Mechanical Guarding*, Bulletin 197, U.S. Dept. of Labor Occupational Safety and Health Administration.
18. F. H. Rolt, *Gauges and Fine Measurements*, University Microfilms.
19. H. J. J. Braddick, *The Physics of Experimental Methods*, Chapman & Hall.
20. T. N. Whitehead, *The Design and Use of Instruments and Accurate Mechanisms*, Dover Publications.
21. W. Mendenhall, *Statistics for the Engineering and Computer Sciences*, MacMillan.
22. R. Loewy, *Industrial Design*.

23. One of the several biographies on Clarence “Kelly” Johnson, who participated in the design of many of the commercial and military aircraft produced by Lockheed Corp.

Appendix C

DE-ICE Survey

C.1 Web Survey (see Farnworth [17])

The MIT Department of Aeronautics and Astronautics is currently investigating of the next generation of Design Environments for Integrated Concurrent Engineering for learning (DE-ICE) in collaboration with Microsoft Inc. DE-ICE will enable integrated engineering and collaborative communication among a distributed multidiscipline engineering team. The system will be used by MIT undergraduate and graduate students in design classes and extended projects.

This survey will be kept completely anonymous.

This effort will be greatly enhanced if you would take a few minutes to give us your feedback.

1. Please tell us about yourself

<u>Under Graduate Student</u>	0-2	3-5	6-10	10+
Years of university experience	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Years of industry experience	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Age				
If you would like to include the organization that you are associated with please feel free to do so.	<input type="text"/>			
Have you used an integrated engineering environment, if so how much.		<input type="text" value="NA"/>		
What proportion of your time is spent in teams on engineering projects.		<input type="text" value="NA"/>		
Have you had any formal training in team dynamics/ working in teams.		<input type="text" value="NA"/>		
Have you had any training in team problem solving skills		<input type="text" value="NA"/>		

2. What is your preferred Operating System:

3. Importance of design tools:

Please rate the importance of these tools for your work.

Hi = Can't Live Without
 Med = Often Useful
 Low = Rarely Useful

(please select all tools you regularly use)

ProE <input type="text" value="NA"/>	CATIA <input type="text" value="NA"/>	AutoCAD <input type="text" value="NA"/>	IDEAS <input type="text" value="NA"/>	MasterCam <input type="text" value="NA"/>
Matlab <input type="text" value="NA"/>	MathCAD <input type="text" value="NA"/>	Mathematics <input type="text" value="NA"/>	DrawCraft <input type="text" value="NA"/>	Visio <input type="text" value="NA"/>
DOORS <input type="text" value="NA"/>	RDD100 <input type="text" value="NA"/>	VenSim <input type="text" value="NA"/>	ACSL <input type="text" value="NA"/>	MstrnX <input type="text" value="NA"/>
Fortran <input type="text" value="NA"/>	C++ <input type="text" value="NA"/>	ADA <input type="text" value="NA"/>	Visual Basic <input type="text" value="NA"/>	Turbo Pascal <input type="text" value="NA"/>
Interactive Data Language <input type="text" value="NA"/>	Java <input type="text" value="NA"/>	Basic <input type="text" value="NA"/>	ANSYS <input type="text" value="NA"/>	3D Studio <input type="text" value="NA"/>
Net Meeting <input type="text" value="NA"/>	Nastran <input type="text" value="NA"/>	IDS <input type="text" value="NA"/>		
MS Excel <input type="text" value="NA"/>	Lotus 123 <input type="text" value="NA"/>	MS Access <input type="text" value="NA"/>	Primavera <input type="text" value="NA"/>	
MS Office <input type="text" value="NA"/>	MS Project <input type="text" value="NA"/>	Corel Office <input type="text" value="NA"/>	Other <input type="text"/>	

4. Please rate the importance for effectiveness and efficiency of the design process:

	Not Important	Some Importance	Important	Very Important	Critical Importance
Software and System Usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Integrated Tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
System Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Learning Curve for New Tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ability to use Familiar Tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ability to Work Anywhere	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Graphical Capability and Visualization of Data	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Communication with Team Members	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The ability to share data easily	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Preferred Communication methods					
Email	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Phone	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tele-conferencing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Audio/Visual Presentations / Communication	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Face/Face	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fax	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. How do you best communicate ideas and concepts to your team members:

Sketches	NA	Memos	NA	Discussion	NA	Models	NA	Graphs	NA
Prototypes	NA	Briefings	NA	Design Review	NA	Meetings	NA	Phone/Fax	NA
Email	NA	Reports	NA	Presentations	NA	Shared Database	NA	Other	
Informal	NA	Face to Face	NA	By Accident	NA				

6. How do you communicate factual information with coworkers/team members:

Sketches	NA	Memos	NA	Discussion	NA	Models	NA	Graphs	NA
Prototypes	NA	Briefings	NA	Design Review	NA	Meetings	NA	Phone/Fax	NA
Email	NA	Reports	NA	Presentations	NA	Shared Database	NA	Other	
Informal	NA	Face to Face	NA	By Accident	NA				

7. What do you find are the limiting factors of the following:

	Response Time	Lack of Feedback	Difficult to Represent Complex Ideas	Varying Interpretations	Nothing	Other
Sketches	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Memos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Discussion	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Models	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Graphs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Phone	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Email	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Reports	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Presentations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Meetings	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

8. What is important in limiting your ability to share engineering data/information:

	Not Important	Some Importance	Important	Very Important	Critical Importance
Different file formats	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Different operating systems	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Slow / no network	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Very large files	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Undefined system interfaces	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Non-static interfaces	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interdependent data	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Not located with team members	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. What would make you more inclined to using an integrated engineering environment:

	Not Important	Some Importance	Important	Very Important	Critical Importance
If I know the tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ease of use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Online help	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Demonstrations or classes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ability to integrate and access data from other applications	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mandate from upper management	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
End to end product development tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reduce repetitive tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reduce the time needed to develop a design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Improve design quality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Understand what the limits of the underlying models are	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Have a thorough understanding of the underlying models used in the environment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Please add any additional comments that you think would help make an integrated engineering environment better:

C.2 Results

Faculty Preferred Operating System		Faculty Importance of Design Tools				
		Hi	Med	Low	NA	
		CAD/CAM				
Mac	4	ProE	1	1	2	3
NT	3	AutoCAD	1	0	1	5
		CATIA	0	1	0	6
		IDEAS	0	1	0	6
		MasterCam	0	0	0	7
		DrawCraft	0	0	0	7
		Visio	0	0	0	7
		3D Studio				
		Other	Working Model			
		Conferencing				
		Net Meeting	0	1	0	6
		Data Analysis, Visualization				
		ANSYS	1	0	0	6
		ACSL	0	0	0	7
		MatrixX	0	0	0	7
		Interactive Data Language	0	0	0	7
		Databases				
		IDS	0	0	0	7
		Ms Access	0	0	0	7
		Finite Element Analysis				
		Nastran	1	2	0	4
		Mathematics				
		Matlab	2	4	0	1
		Mathematica	0	1	1	5
		MathCAD	0	1	0	6
		Office				
		Ms Office	7	0	0	0
		Corel Office	0	1	0	6
		Programming				
		Fortran	2	1	0	4
		Java	0	0	1	6
		ADA	0	0	0	7
		Visual Basic	0	0	0	7
		Turbo Pascal	0	0	0	7
		Basic	0	0	0	7
		C+/C++				
		Project Management				
		MS Project	0	1	3	3
		VenSim	0	0	0	7
		Primavera	0	0	0	7
		Requirement Management Tools				
		DOORS	0	0	0	7
		RDD100				
		Spreadsheets				
		MS Excel	3	3	1	0
		Lotus 123	0	0	0	7

Undergrad Preferred Operating System		Undergrad Importance of Design Tools																
		CAD/CAM				Requirements Management Tools				Spreadsheets				Other				
			Hi	Med	Low	NA			Hi	Med	Low	NA			Hi	Med	Low	NA
Unix	6	ProE	1	4	1	12	DOORS	0	0	1	17							
NT	4	AutoCAD	1	2	1	14	RDD100											
Win95	3	IDEAS	1	2	1	14												
Win98	2	CATIA	0	1	1	16	MS Excel	7	6	2	3							
Mac	2	MasterCam	0	0	1	17	Lotus 123	0	0	3	15							
NA	1	DrawCraft	0	0	1	17												
		Visio	0	0	1	17												
		3D Studio																
		Conferencing																
		Net Meeting	1	0	2	15												
		Data Analysis, Visualization																
		Interactive Data Language	0	1	1	16												
		ANSYS	0	0	2	16												
		ACSL	0	0	1	17												
		MatrixX	0	0	1	17												
		Databases																
		MS Access	0	2	2	14												
		IDS	0	0	1	17												
		Other					Filemaker Pro											
		Finite Element Analysis																
		Nastran	0	0	1	17												
		Graphics Presentation																
		Other					MS Power point											
		Mathematics																
		Matlab	9	5	0	4												
		MathCAD	1	0	1	16												
		Mathematica	0	1	2	15												
		Office																
		MS Office	8	4	1	5												
		Corel Office	2	0	2	14												
		Other					emacs-High											
		Orbital Analysis																
		Other					STK High											
		Programming																
		Basic	1	1	3	13												
		Java	0	2	2	14												
		Turbo Pascal	0	1	4	13												
		ADA	0	1	1	16												
		Fortran	0	0	3	15												
		C+/C++																
		Visual Basic																
		Project Management																
		VenSim	1	0	0	17												
		MS Project	0	2	2	14												
		Primavera	0	0	2	16												

Grad Preferred Operating System		Grad Importance of Design Tools										
		CAD/CAM				Project Management				Requirements Management Tools		
			Hi	Med	Low	NA		Hi	Med	Low	NA	
NT	10	Visio	1	3	3	19	MS Project	1	2	4	19	
Unix	7	ProE	0	5	6	15	VenSim	0	1	1	24	
Win95	4	IDEAS	0	4	1	21	Primavera	0	0	1	25	
Mac	3	AutoCAD	0	3	4	19	Requirements Management Tools					
Win98	1	MasterCam	0	0	4	22	DOORS	1	0	1	24	
NA	1	CATIA	0	0	2	24	RDD100					
		DrawCraft	0	0	1	25	Spreadsheets					
		3D Studio					MS Excel	13	9	2	2	
		Other	POV-Ray					Lotus 123	0	1	3	22
Conferencing							Other					
		Net Meeting	0	1	4	21	all Adobe products, Regular network connectivity: Email, ssh, www browser clients					
Data Analysis, Visualization												
		ANSYS	3	0	3	20						
		ACSL	1	1	0	24						
		Interactive Data Language	0	1	1	24						
		MatrixX	0	0	2	24						
		Other	Labview									
Databases												
		MS Access	1	2	3	20						
		IDS	0	0	2	24						
Finite Element Analysis												
		Nastran	2	1	1	22						
		Other	ABAQUS									
Graphics Presentation												
		Other	Powerpoint									
Mathematics												
		Matlab	21	3	1	1						
		Mathematica	3	3	3	17						
		MathCAD	2	1	5	18						
		Other	Maple, Maple Med									
Office												
		MS Office	18	7	0	1						
		Corel Office	0	1	3	22						
		Other	FrameMaker, LaTeX									
Photo Editing												
		Other	Photoshop									
Programming												
		Java	2	3	3	18						
		Fortran	2	0	6	18						
		Turbo Pascal	0	0	4	22						
		Basic	0	0	4	22						
		ADA	0	0	2	24						
		C+/C++										
		Visual Basic										

Faculty Design Process Effectiveness and Efficiency

	Critical Importance	Very Important	Important	Some Importance	Not Important
Software and System Usability	2	3	1	0	0
Graphical Capability and Visualization of Data	1	2	3	0	0
Learning Curve for New Tools	1	2	2	1	0
Ability to use Familiar Tools	1	2	1	2	0
Ability to Work Anywhere	1	1	3	0	1
System Speed	1	0	4	1	0
The ability to share data easily	0	3	3	0	0
Integrated Tools	0	3	1	2	0
Communication with Team Members	0	2	4	0	0

Preferred Communication Methods

	Critical Importance	Very Important	Important	Some Importance	Not Important
Face/Face	5	1	0	0	0
Email	4	2	1	0	0
Audio/Visual Presentations/Communication	2	3	1	1	0
Phone	1	1	4	1	0
Fax	1	1	0	4	1
Tele-conferencing	0	1	2	4	0

Faculty Best Communication of Ideas and Concepts					Faculty Communication of Factual Information				
	Hi	Med	Low	NA		Hi	Med	Low	NA
Meetings	5	1	0	1	Email	6	1	0	0
Email	4	2	1	0	Meetings	5	1	0	1
Sketches	4	0	1	2	Presentations	4	1	2	0
Presentations	2	4	1	0	Models	3	2	1	1
Models	2	3	1	1	Briefings	3	2	0	2
Prototypes	2	2	1	2	Prototypes	2	1	2	2
Briefings	2	2	1	2	Graphs	2	0	0	5
Phone/FAX	1	2	1	3	Memos	1	2	3	1
Memos	0	3	3	1	Phone	1	0	1	5
Discussion					Sketches	0	2	1	4
Graphs					Discussion				
Design Review					Design Review				
Reports					Reports				
Shared Database					Shared Database				
Informal					Other				
Face to Face					Informal				
By Accident					Face to Face				
Other					By Accident				

Undergrad Design Process Effectiveness and Efficiency

	Critically Important	Very Important	Important	Some Importance	Not Important
Communication with Team Members	8	4	2	2	0
Software and System Usability	7	4	5	0	0
Ability to Work anywhere	5	4	4	3	0
Learning Curve for New Tools	3	4	5	3	1
System Speed	2	5	4	5	0
Integrated Tools	2	5	4	4	1
Graphical Capability and Visualization of Data	3	3	7	3	0
The ability to share data easily	2	10	4	0	0
Ability to use Familiar Tools	0	5	6	4	1

Preferred Communication Methods

	Critically Important	Very Important	Important	Some Importance	Not Important
Face/Face	10	4	2	0	0
Email	7	5	4	0	0
Phone	3	9	2	0	2
Audio/Visual Presentations/Communication	3	2	5	3	2
Tele-conferencing	0	6	3	1	5
Fax	0	2	5	3	4

Undergrad Best Communication Ideas and Concepts					Undergrad Communication of Factual Info				
	Hi	Med	Low	NA		Hi	Med	Low	NA
Email	11	3	1	3	Email	13	3	0	2
Sketches	7	6	0	5	Meetings	5	5	1	7
Meetings	5	7	2	4	Sketches	4	3	2	9
Presentations	2	7	5	4	Presentations	3	6	2	7
Models	1	7	1	9	Briefings	3	2	2	11
Briefings	1	5	2	10	Memos	2	3	4	9
Phone/FAX	1	4	1	12	Prototypes	1	1	5	11
Memos	1	1	8	8	Models	1	1	4	12
Prototypes	0	4	3	11	Phone	0	4	1	13
Discussion					Graphs	0	3	0	15
Graphs					Discussion				
Design Review					Design Review				
Reports					Reports				
Shared Database					Shared Database				
Informal					Other				
Face to Face					Informal				
By Accident					Face to Face				
Other					By Accident				

Grad Design Process Effectiveness and Efficiency

	Critically Important	Very Important	Important	Some Importance	Not Important
The ability to share data easily	9	11	6	0	0
Communication with Team Members	9	8	8	1	0
Software and System Usability	8	16	1	1	0
Graphical Capability and Visualization of Data	5	11	7	3	0
System Speed	4	15	6	1	0
Learning Curve for New Tools	3	11	7	5	0
Integrated Tools	3	9	9	5	0
Ability to use Familiar Tools	2	11	9	3	1
Ability to Work anywhere	1	9	8	7	0

Preferred Communication Methods

	Critically Important	Very Important	Important	Some Importance	Not Important
Face/Face	14	8	4	0	0
Email	13	9	3	1	0
Phone	3	9	10	3	1
Audio/Visual Presentations/Communication	2	5	16	3	0
Tele-conferencing	1	2	4	15	3
Fax	0	2	6	6	12

Grad Best Communication of Ideas and Concepts

Grad Communication of Factual Information

	Hi	Med	Low	NA		Hi	Med	Low	NA
Sketches	14	9	1	2	Meetings	14	8	1	3
Email	13	8	3	2	Email	13	7	1	5
Meetings	10	13	3	0	Presentations	11	9	2	4
Presentations	7	12	5	2	Graphs	10	4	0	12
Briefings	4	9	3	10	Briefings	9	9	1	7
Models	3	10	8	5	Sketches	4	6	6	10
Prototypes	3	5	7	11	Models	4	6	6	10
Phone/FAX	1	6	7	12	Memos	3	8	6	9
Memos	0	12	7	7	Prototypes	3	5	5	13
Discussion					Phone	0	3	8	15
Graphs					Discussion				
Design Review					Design Review				
Reports					Reports				
Shared Database					Shared Database				
Informal					Other				
Face to Face					Informal				
By Accident					Face to Face				
Other					By Accident				

Faculty Limiting Factors

Sketches	Varying Interpretations	Lack of Feedback	Difficult To Represent	Nothing	Other	Response Time
	2	1	1	1	1	0
Memos	Varying Interpretations	Response Time	Lack of Feedback	Difficult To Represent	Time	Nothing
	2	1	1	1	1	0
Discussion	Difficult To Represent	Varying Interpretations	Nothing	Can't save/database ideas, concepts	Response Time	Lack of Feedback
	2	1	1	1	0	0
Models	Difficult To Represent	Response Time	Varying Interpretations	Nothing	Lack of Feedback	Other
	2	1	1	1	0	0
Graphs	Nothing	Lack of Feedback	Response Time	Difficult To Represent	Varying Interpretations	Other
	4	1	0	0	0	0
Phone	Response Time	Difficult To Represent	Nothing	Lack of Feedback	Varying Interpretations	Other
	2	2	2	0	0	0
Email	Difficult To Represent	Nothing	Lack of Feedback	Response Time	Varying Interpretations	Other
	3	2	1	0	0	0
Reports	Response Time	Lack of Feedback	Nothing	Time	Difficult To Represent	Varying Interpretations
	3	1	1	1	0	0
Presentation	Nothing	Time, can't save ideas	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations
	3	2	1	0	0	0
Meetings	Response Time	Nothing	Difficult To Represent	Can't save ideas	Lack of Feedback	Varying Interpretations
	2	2	1	1	0	0
Other	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations	Nothing	Other

Undergrad Limiting Factors

Sketches	Difficult To Represent	Varying Interpretations	Drawings Skill, Accuracy	Lack of Feedback	Nothing	Response Time
	5	4	2	1	1	0
Memos	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations	Nothing	Other
	5	3	2	2	1	0
Discussion	Difficult To Represent	Varying Interpretations	Nothing	Accuracy, Time limit	Response Time	Lack of Feedback
	3	3	3	2	1	0
Models	Response Time	Difficult To Represent	Nothing	Lack of Feedback	Varying Interpretations	Other
	5	3	2	1	1	0
Graphs	Varying Interpretations	Difficult To Represent	Nothing	Lack of Feedback	Response Time	Other
	6	3	2	1	0	0
Phone	Difficult To Represent	Response Time	Varying Interpretations	Lack of Feedback	Nothing	Other
	7	5	1	0	0	0
Email	Difficult To Represent	Response Time	Asynchronous Comm	Lack of Feedback	Varying Interpretations	Nothing
	7	2	2	1	1	0
Reports	Response Time	Lack of Feedback	Varying Interpretations	Difficult To Represent	Nothing	Other
	8	3	1	0	0	0
Presentation	Nothing	Response Time	Time, Drawing Skill	Lack of Feedback	Varying Interpretations	Difficult To Represent
	4	3	2	1	1	0
Meetings	Nothing	Difficult To Represent	Varying Interpretations	Other	Response Time	Lack of Feedback
	6	2	2	1	0	0
Other	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations	Nothing	Other

Grad Limiting Factors

Sketches	Varying Interpretations	Difficult To Represent	Nothing	Ease of Distribution	Response Time	Lack of Feedback
	12	9	2	1	0	0
Memos	Lack of Feedback	Difficult To Represent	Response Time	Varying Interpretations	Nothing	Other
	11	5	4	2	2	0
Discussion	Nothing	Difficult To Represent	Varying Interpretations	Response Time	Lack of Feedback	Other
	14	4	4	1	1	0
Models	Response Time	Nothing	Difficult To Represent	Varying Interpretations	Lack of Feedback	Time, Ease of Distribution
	9	6	3	2	1	2
Graphs	Nothing	Difficult To Represent	Varying Interpretations	Response Time	Lack of Feedback	Other
	8	7	4	1	1	0
Phone	Difficult To Represent	Nothing	Varying Interpretations	Response Time	Ease of Distribution	Lack of Feedback
	15	4	2	1	1	0
Email	Difficult To Represent	Nothing	Varying Interpretations	Response Time	Lack of Feedback	Other
	8	8	4	3	1	0
Reports	Response Time	Lack of Feedback	Time to create, Efficiency(time/payoff)	Nothing	Difficult To Represent	Varying Interpretations
	9	7	4	2	0	0
Presentation	Nothing	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations	Time to create
	7	3	3	3	3	3
Meetings	Nothing	Varying Interpretations	Difficult To Represent	Time, poor meeting skills	Response Time	Lack of Feedback
	9	4	3	3	2	1
Other	Response Time	Lack of Feedback	Difficult To Represent	Varying Interpretations	Nothing	Other

Faculty Limits in Ability to Share Engineering Data/Information

	Critical Importance	Very Important	Important	Some Importance	Not Important
Other	2	1	1	0	1
Different File Formats	2	0	3	1	0
Slow/No Network	1	1	3	1	0
Different Operating Systems	1	1	2	2	0
Undefined System Interfaces	1	1	1	2	0
Non-static Interfaces	1	0	1	1	2
Very Large Files	0	1	3	2	0
Not Located With Team Members	0	0	1	2	1

Faculty Inclined To Use And Integrated Engineering Environment

	Critical Importance	Very Important	Important	Some Importance	Not Important
Understand what the limits of the underlying models are	4	2	0	0	0
Have a thorough understanding of the underlying models used in the environment	4	0	1	1	0
Reduce time needed to develop a design	3	3	0	0	0
Improve design quality	3	3	0	0	0
Ease of Use	2	3	1	0	0
Reduce repetitive tasks	2	2	1	1	0
If I know the Tools	2	1	2	1	0
Ability to Integrate and Access Data	1	2	2	1	0
Online Help	1	2	1	2	0
Demonstrations or Classes	1	0	2	1	0
End to end product development tools	0	4	1	1	0
Mandate from upper Management	0	3	2	0	1

Faculty Comments

Key issue is that the process, particularly embedded models and databases are transparent to all users - this requires careful documentation and establishing a discipline in its use. Major weakness in my experience is that some users do not understand the quality and fidelity of the data or models that they are utilizing

Undergrad Limits in Ability to Share Engineering Data/Information

	Critical Importance	Very Important	Important	Some Importance	Not Important
Slow/No Network	6	2	2	2	0
Not Located With Team Members	4	1	4	2	1
Very Large Files	3	3	3	3	1
Undefined System Interfaces	3	2	3	1	2
Other	1	6	3	0	2
Different Operating Systems	1	4	3	4	1
Different File Formats	1	3	4	3	1
Non-static Interfaces	0	3	4	1	2

Undergrad Inclined To Use And Integrated Engineering Environment

	Critical Importance	Very Important	Important	Some Importance	Not Important
Reduce repetitive tasks	5	5	2	1	0
Reduce time needed to develop a design	5	5	2	1	0
Understand what the limits of the underlying models are	4	4	3	1	1
Improve design quality	3	9	1	0	0
If I know the Tools	3	6	4	0	2
Ease of Use	3	4	6	2	0
Ability to Integrate and Access Data	2	3	6	3	0
Online Help	2	3	4	4	1
End to end product development tools	2	3	2	4	2
Demonstrations or Classes	2	1	5	5	1
Have a thorough understanding of the underlying models used in the environment	1	6	4	3	1
Mandate from upper Management	1	4	5	1	3
Data that shows these tools help	0	1	0	0	0

Undergrad Comments

Hi Bruce,
I am only filling in things as a test, since I have just received email from some who says that this survey did not work.
Phyllis

Hi Bruce,
This time I am taking your survey on my Windows NT machine (previously I used my MAC G3). So far it seems to be working on this platform. I will send email to the complainant to see what platform he/she was using.
Phyllis Collymore (pco)

Recognition that integrated engineering is not always the way to go. Integrated is another word for complex, which can give way to chaos. I often find when working in a group that people tend to let the team (or the software) do their engineering for them, and that doesn't work. If this survey has any implications for academics, which is unclear, then I would make the point that the practice of integrated engineering should be a secondary skill to fundamental engineering knowledge. Free Body diagrams are much more important than excel, and the de = dq - dw can take AutoCAD any day of the week and twice on sunday.

Grad Limits in Ability to Share Engineering Data/Information

	Critical Importance	Very Important	Important	Some Importance	Not Important
Slow/No Network	6	7	7	2	2
Different File Formats	4	11	4	5	0
Undefined System Interfaces	4	4	7	5	4
Different Operating Systems	2	11	6	4	1
Other	1	6	7	5	5
Not Located With Team Members	1	3	9	6	5
Non-static Interfaces	1	3	6	6	4
Very Large Files	0	8	6	8	2

Grad Inclined To Use And Integrated Engineering Environment

	Critical Importance	Very Important	Important	Some Importance	Not Important
Ability to Integrate and Access Data	9	10	5	1	0
Ease of Use	8	12	3	2	0
Improve design quality	7	8	8	1	0
Reduce time needed to develop a design	6	9	7	2	0
Understand what the limits of the underlying models are	5	6	12	1	0
Have a thorough understanding of the underlying models used in the environment	5	6	8	4	0
If I know the Tools	4	9	8	2	2
Reduce repetitive tasks	3	13	6	3	0
Demonstrations or Classes	2	5	11	6	1
Mandate from upper Management	2	3	6	9	5
Online Help	1	7	10	5	2
End to end product development tools	1	5	9	6	4
Well defined process in place	1	0	0	0	0

Grad Comments

this is a very good survey

After reading this entire survey, I still have little idea what an "Integrated Design Environment" is. It sounds like some buzzword thought up by business majors or LAI folk to impress shareholders. Your first task is to relay to the students exactly WHAT DE-ICE is.

From my experience with 'integrated' environments in a class setting I have observed:

- a) Without a well designed and understood process for collaboration, much effort is spent trying to figure out this mechanism on-the-fly. The amount of actual design goes through the floor. Assuming such a process exists, maybe an IAP class in preparation for a spring design class would be the best way to introduce it.
- b) In a class setting (limited time, etc) distance 'education' always provides a mediocre experience. A competent lecturer in front of you is better than a genius on a TV screen. For design classes, the extra effort that must be put into coordination detracts from that that could go into design.

Please, please, please do not convert all work space to open rooms without private work areas! While being in close proximity is good for some amount of time, being together for all of the work time reduces productivity. I have seen many different computer facilities and work environments during my career to date, and the open environments foster distractions and lower the ability to concentrate. We work on difficult problems in aerospace, and we need to be able to think and focus well on our problems. While it may seem like a total integration of resources and people is "good for the team", I strongly believe that a group environment is not always good for the work. I do know that others feel the same way - just ask around and see who can concentrate well in a full Athena cluster where people are talking.

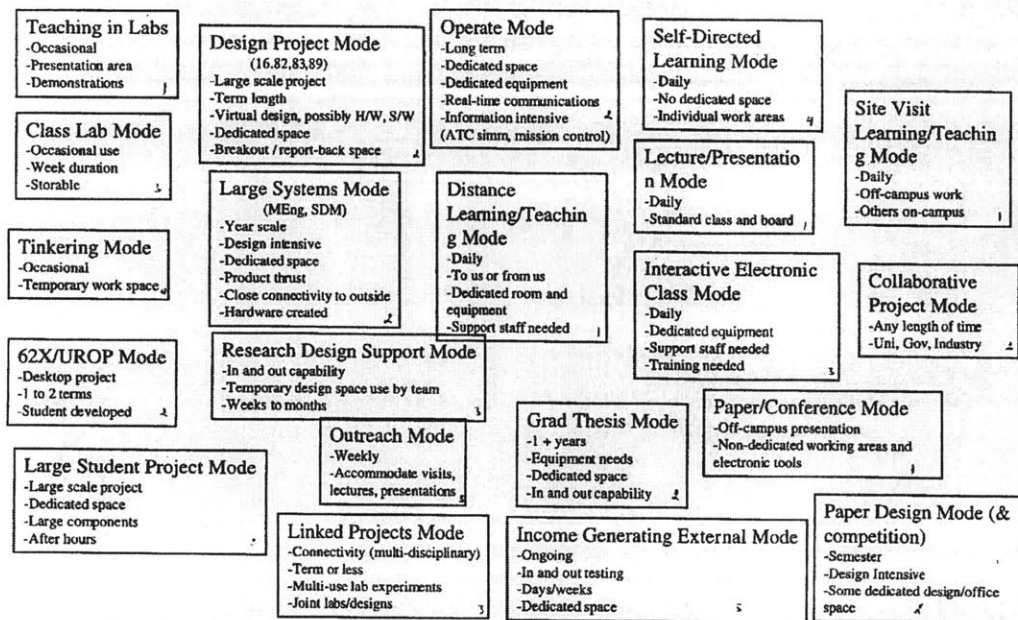
Who wrote this survey? It's way too abstract for my practical engineering tastes. "Non-static interfaces"? "Understand what the limits of the underlying models are"? What are we talking about here? You need more concrete statements.

Appendix D

MIT Operational Modes

Operation Modes

• We have modes that are either physical equipment-based, paper design-based, or a combination of both. The list of possible department operating modes is given below



How We Operate

The department of Aeronautics and Astronautics foresees itself operating in many types of educational modes. Some of these modes are classroom/theory focused, while others are more hands-on/design focused. The following list of modes covers how the department might function:

1. Large Systems Mode (M.Eng, LFM, SDM)
2. Design Project Mode (16.82,.83,.89)¹
3. 16.62X/UROP Project Mode
4. Large Student Project Mode
5. Class Lab/Experiment Mode
6. Operate Mode
7. Linked Projects Mode
8. Grad Thesis Mode
9. Teaching in Labs Mode
10. Research Design Support Mode
11. Income Generating External Mode
12. Outreach Mode
13. Tinkering Mode
14. Self-Directed Learning Mode
15. Lecture/Presentation Mode
16. Interactive Electronic Class Mode
17. Paper/Conference Mode
18. Paper Design Mode (& competition)
19. Collaborative Project Mode
20. Site Visit Learning/Teaching Mode
21. Distance Learning/Teaching Mode

¹ Aero/Astro Courses, 16.82-Flight Vehicle Engineering, 16.83-Space Systems Engineering, 16.89-Space Systems Engineering

Operational Modes Defined

The following sections provide definitions of the department's operational modes.

Large Systems Mode (M.Eng, LFM, SDM)

The Large Systems Mode represents programs that are very design intensive and team oriented, requiring a certain amount of dedicated space for doing the design work and systems prototyping for periods of time ranging from a semester to a year. The definition of large system is somewhat ambiguous, but will be taken to mean a system that requires several disciplines to create it, and results in a final product or prototype consisting of varying amounts of H/W, S/W, and L/W. The design work covers paper/virtual design and some levels of physical prototyping and testing.

Design Project Mode (16.82, .83, .89)

The Design Project Mode represents class-based design projects that are carried out over the period of a semester by student teams from a given class. The work includes engineering trades from various facets of the field of aerospace engineering, perhaps trades from socio-economic and environmental issues, some computer simulation and visualization, and results in a paper design. The work may be conducted by 1 large class team (10-20 people), or several smaller teams (3-8 people).

16.62X/UROP Project Mode

This mode focuses on the department's senior research project which involves teams of two undergraduate students designing, building, operating, and reporting on an experiment that is conducted under the guidance of a faculty advisor. This mode is conducted over two semesters, where the first semester is dedicated to the background research, and the second is dedicated to building the apparatus (typically desktop size), running the experiment, and reporting the results to the faculty in a formal presentation and document.

Large Student Project Mode

The Large Student Project Mode centers on projects that are typically extra-curricular, but are geared towards building something that is used in a competition (Human Powered Aircraft, Robotic Helicopter, Solar Car). The teams working on these projects are from several engineering disciplines and require a dedicated office for running the project, design space, building and testing space, storage space, and access to the facilities anytime that is convenient for them (this is typically after hours, since the project is extra-curricular). These year-long to multi-year projects involve teams of 5 to 20 people and result in operational prototypes of significant size (tens of feet in length and tens to hundreds of pounds in weight).

Class Lab/Experiment Mode

The Class Lab/Experiment Mode is the traditional mode that labs operate in where the instructor has assigned a lab for the class to perform, collect and reduce data, and write a report on the procedure. These labs are usually conducted by small teams from the class (3-6 students), and require bench-top set-ups, or the use of the department's wind tunnels.

Operate Mode

The Operate Mode is intended to teach the students about the operational concepts of engineering systems, and give them hands-on experience in the operation of the

systems. This mode requires people dedicated to the maintenance and operation of the systems, long-term dedicated space for the equipment, and probably real-time communications with other departments/sites for systems such as satellite base stations.

Linked Projects Mode

The Linked Projects Mode represents term to year-long projects that are set-up as interdisciplinary projects between several section of the department and/or the school. The projects can be either multi-use projects that serve as labs for several classes (i.e. wing spar for structures, airfoil for aerodynamics, aileron for control and stability -> all resulting in a wing design) or a joint project between disciplines, where the team is formed of various student show work together, but do the work that is centered on their discipline (i.e. wing design by two Aero students, programmed logic circuit built and installed by EECS student to enact control laws). The interdisciplinary nature of this mode will require meeting space, work space, storage space, and possibly formal presentation space.

Grad Thesis Mode

The Grad Thesis Mode is intended to support a graduate student who needs to establish an experimental set-up for some period of time, that cannot be supported by their current research lab. The time scale could be from the semester length, to several years, requiring that the space be dedicated for that amount of time.

Teaching in Labs Mode

The Teaching in Labs Mode is where faculty will actually bring their classes to the lab to demonstrate certain principles and phenomena with equipment that is specific to the lab (wind tunnels, space truss, etc.). An extension of this would be to use any conference-style or electronic classrooms as teaching rooms in order to maximize the space use, and/or utilize equipment specific to the room (i.e. networked projection units, video equipment, etc.).

Research Design Support Mode

The design-center capabilities of the labs would be used by research teams to come in for a short period of time (hours/days) to work through a segment of the research design. This would be a short-term dedication of space that would support the research team's design efforts with analysis tools, design tools, communications, and presentation equipment. This mode may be directly supported by the distance learning mode, where the communications equipment could be used for meetings with research sponsors.

Income Generating External Mode

This mode will support external companies who lease the use of facilities such as the Wright Brother's Wind Tunnel. This mode typically lasts for several weeks, and requires the dedicated use of the equipment, support staff to operate the equipment, and some space. Security of information may be an issue with some companies.

Outreach Mode

This mode is used to support the publicity and awareness of the department and the new teaching labs. Tours of the university will pass through the new teaching labs and be given an introduction to what happens in the Aero/Astro department. An extension of this might be sending people from the department out to talk about the department and labs.

Tinkering Mode

The Tinkering Mode represents the individual projects people do on their spare time, that typically require the use of shop equipment (building models, furniture, repairing

a bike or toaster, etc.). This mode needs some shop equipment, tools, and work surfaces, and happens any time of day when the shop is open.

Self-Directed Learning Mode

This mode is where students go and learn material on their own. Traditionally this has meant reading text books and going to the library, but with the advent of new communications technology, this may include borrowing educational videos to watch, surfing the Internet for information and programs, or exploring any other media or material that will help the students gain a better understanding of the subject they are learning about.

Lecture/Presentation Mode

This is the standard class mode, where a lecturer discusses/presents class material. While traditionally performed on chalk boards and overheads, some faculty are now moving to Web-based courses that require electronic presentation. H/W and S/W in the classroom for showing the course material, videos, and running simulations to demonstrate theories and principles.

Interactive Electronic Class Mode

This class mode represents the fully electronic classroom where students are able to do computer-based work in real-time, under the tutelage of faculty, using interactive S/W to mark/grade/comment on work and projection equipment to demonstrate certain work. This may be extended to include the interactive design class where students perform trade studies and design work on a large project.

Paper/Conference Mode

This mode represents the typical research paper/presentation mode that many graduate students experience in their graduate career. This mode usually occurs off campus and involves simple overhead, slide, or video presentations.

Paper Design Mode (& competition)

This mode represents design competition work where the end result is a document and presentation that is used as the material for a competition. The support needed for such a mode includes design tools, management tools, visualization tools, etc. This mode stays in the electronic and paper realm and does not result in the construction of any H/W.

Collaborative Project Mode

This mode represents the department working on projects in collaboration with other universities, government, and industry. The project may be conducted as a response to one of these other organizations needs (similar to a contract), or in partnership, where the teams members are all working on various segments of the system. This mode is communications intensive, and requires data, voice, and video communications, often in real time.

Site Visit Learning/Teaching Mode

This mode can operate in two ways. The first is students going out to visit sites that have tools, equipment, etc. that are of interest to the subject being taught, and the course is taught on site for the given day to let students see the application of the theory in real life, including the design, manufacturing, implementation and operation of some of the systems. The second way this could operate, is by the students acting as "teacher-hosts" for part of a day, where a sponsoring company might send a group of employees to learn about what the students are doing in their research, etc. This mode allows the students to teach the material, which reinforces their learning, and also allows the interaction of

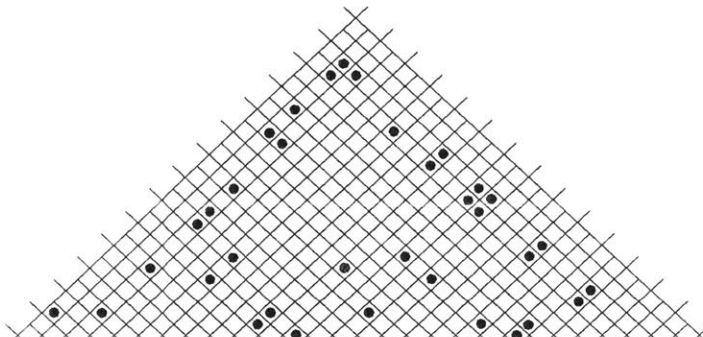
industry and students to bring to light the application and operation of many of the ideas they are working with at MIT.

Distance Learning/Teaching Mode

This mode represents the video-conference classroom, where the department is broadcasting classes/presentations to other sites, or receiving classes/presentations at this site.

Appendix E

Complete DE-ICE QFD Matrix

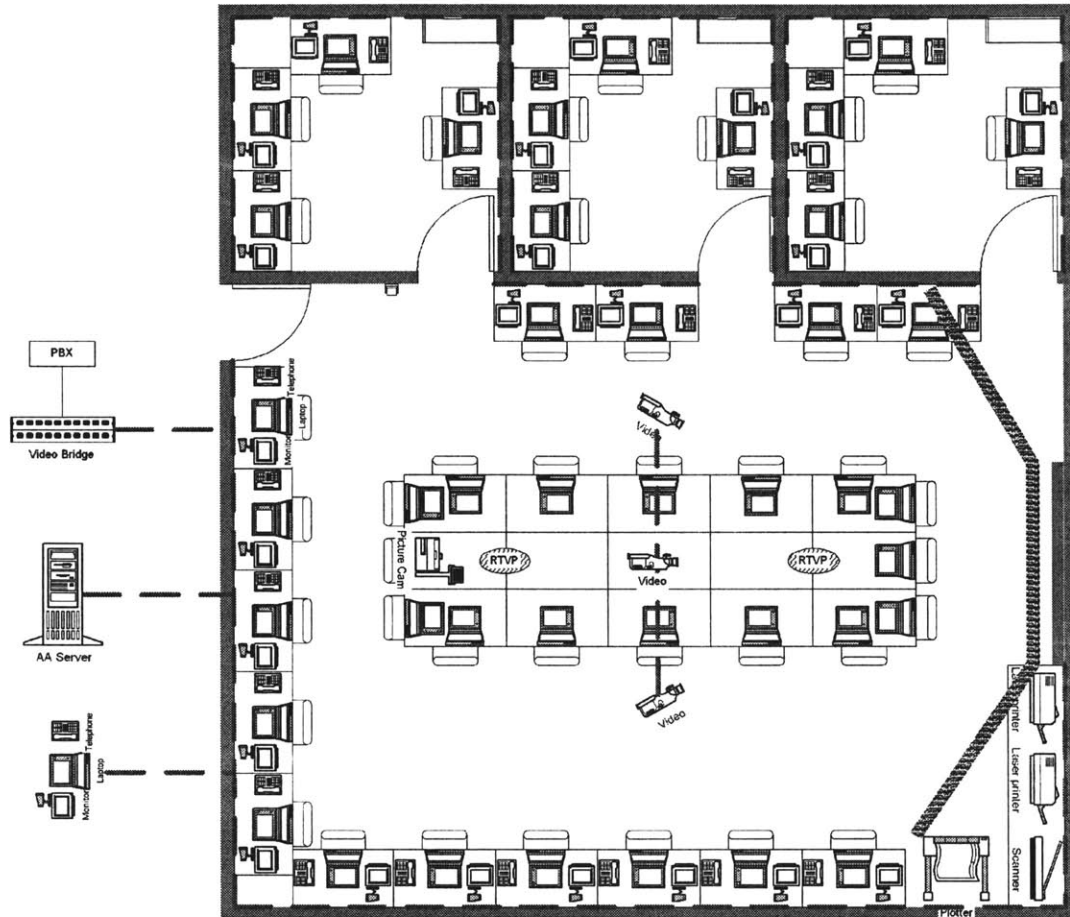


High Level Needs	Technical Requirements	Customer Needs																															
	Customer Needs	Rating Factor	Provide Guidance throughout design process	Interactive course delivery support	Design and analysis support	Creative thinking support	Education/Training support	Manage perceived complexity of the design	Information, knowledge and expert help	Manufacturing process capability support	Life cycle cost support	Visualization of design	Planning and management of design process	Rapid exploration of design solutions	Optimize collaborative support	Integration and management of data	Familiar tools	Reduce repetitive tasks	Learning curve	Supports team building	Utilize COTS software	Documentation of the DE-ICE system design	Flexible Systems	Low Upgrade Costs	Low operating costs	Maintainable System	Follow recognized standards	Access to system from any location	Optimal on any platform	System available 24/7	Personalization and reporting support		
Pedagogical	Active Learning in a Design Environment	10	9	9	9	9	9	9	1	1																							64
	Holistic View of Design	9	3	9	9							1	3	3																			31
	Improved Knowledge of and Experience with the Design Process	8	3	1	3	1	3	1	9					1	1	3																	
Operational	Support of Life Cycle Analysis	5		3						9	9																						28
	Improved Quality of Student Design Work	8	1	3	3	3	1	3				3		3		1	1	3															27
	Increased Productivity for Given Amount of Time	7	3	9		1	3	3				1	9	9	9	9	9	9	3	1	1	1											108
	Support For Team Enablers	7											3			1																	6
	Sustainable System	6															1					9	9	9	9	9	9	3					61
Highly Usable system	4															1	3	9	3	9	3	3	3									58	
Constraint	Capability to Support MIT Operational Modes	N/A	1	3	9	3	3	3																									74
	TR Importance Weight	173	96	297	122	165	128	127		45	54		68	149	132	122	105	113	117	107	130	73	98	60	64	64	51	45	71	68	72	32	
	Relative Weight (%)	2	4	1	3	2	3	5		5	5		2	3	3	3	3	3	3	3	5	5	4	5	4	5	7	5	5	5	5		
	Units	# number of types of devices available	X sec latency	# setup needs to create and analyze	# methods available	# of types of communication available	# of positions under design	Time to access	# processes supported	methods of costing support	uncertainty estimates	X sec to render visual	# Optimization Methods	# concepts explored/hr	# users, FPS, latency	turnover time < X min	% common tools in the system	% time non reproducible	< X hr	% time working as a team	% COTS	docs include manual	# docs in system, X time to install/docs	< X \$	< X \$	# repair / month	ISO 9000	time to access content	number of platforms supported	% time system up and running	Time to program days		
	Target Values																																
Benchmarking	Baob	>1	8	1	>4			>1 min				>2		>4	<1																		
	NASA, Aerospace, TRW	>2	7	0								1		0																			2
	CalTech	>4	7	0								1		1																			1
	COMET	>2	4	1								2		inf																			3
	IMAGE																																

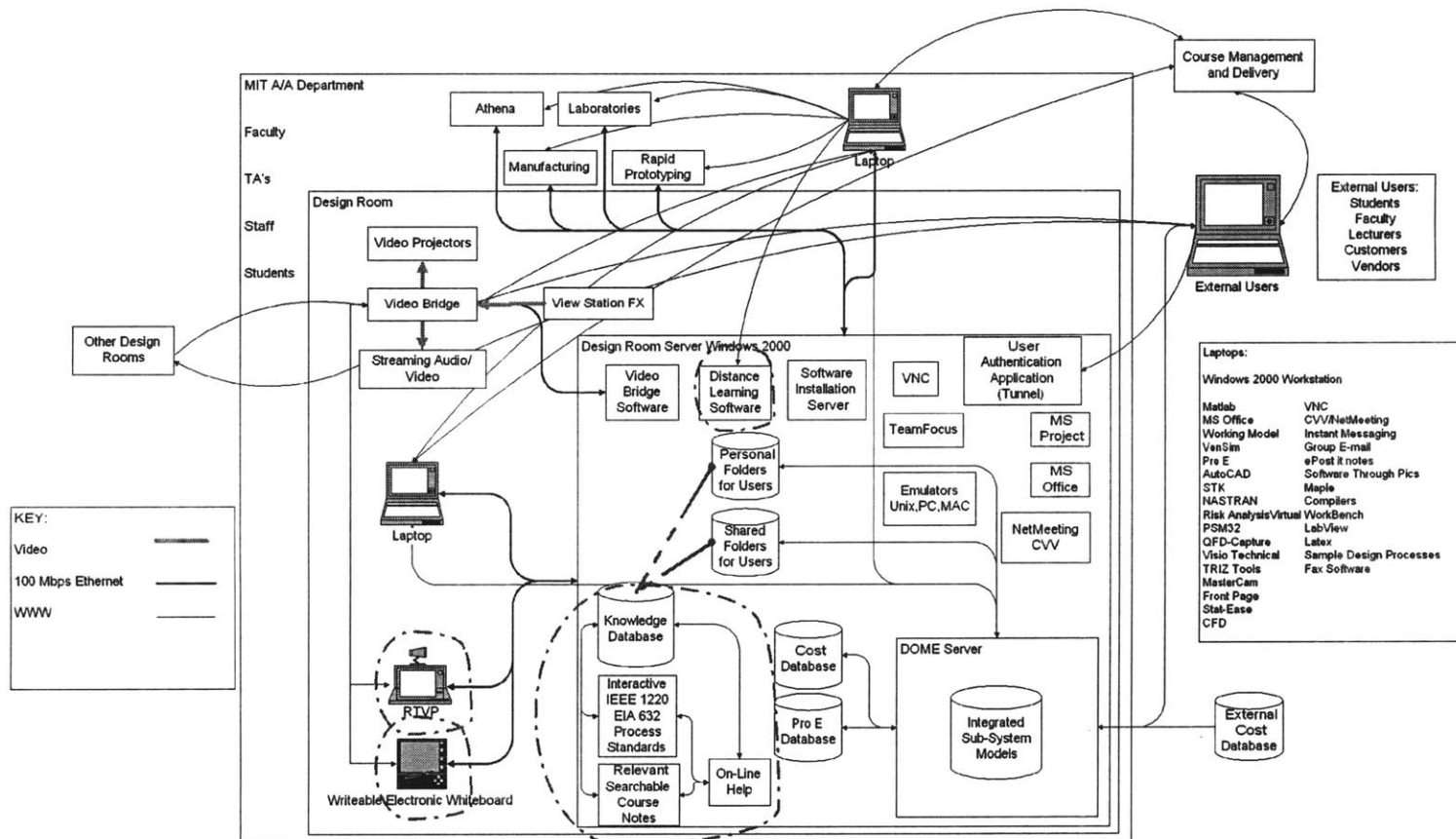
Appendix F

Architecture Variants

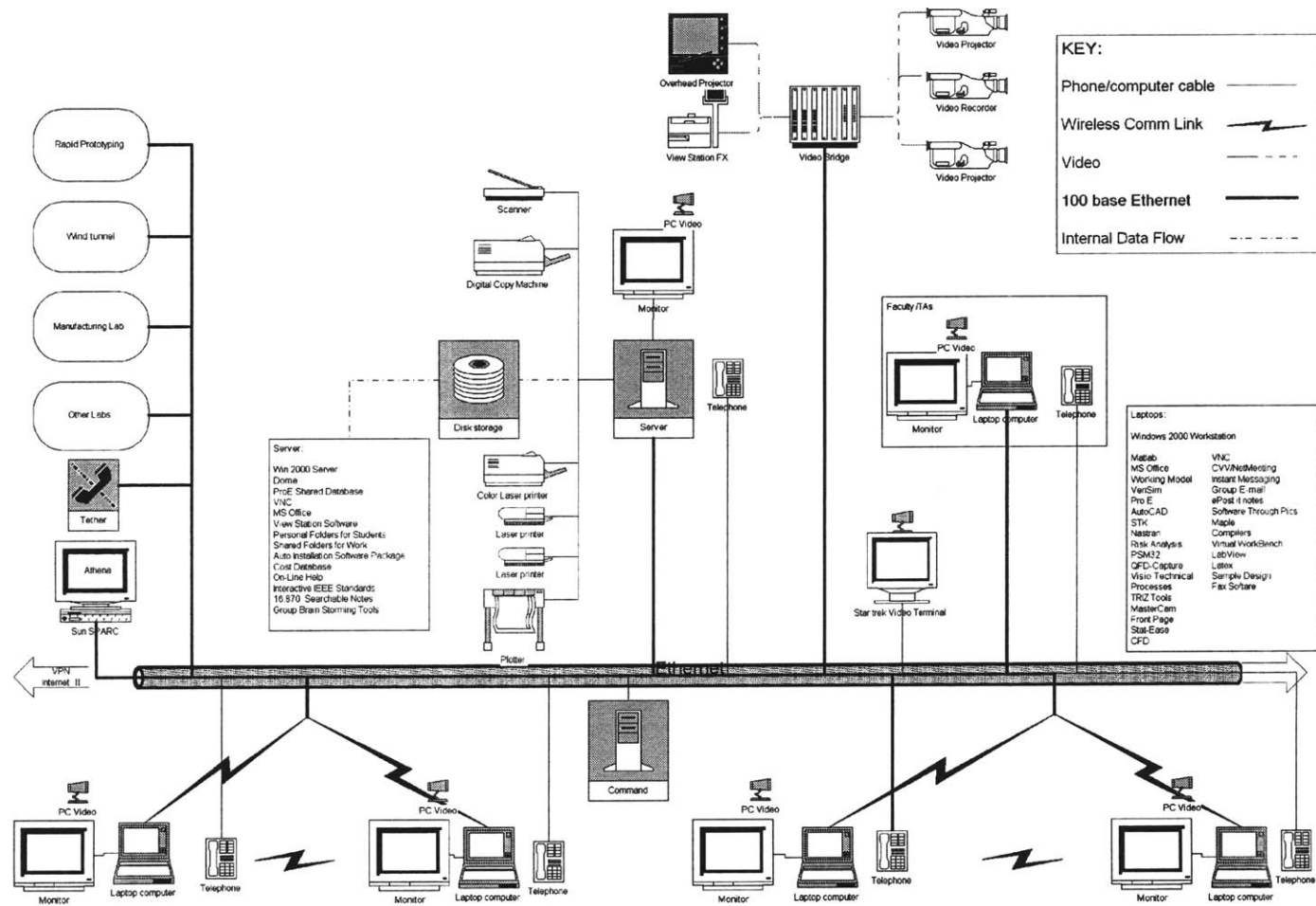
F.1 Architecture A Physical View (see Farnworth [17])



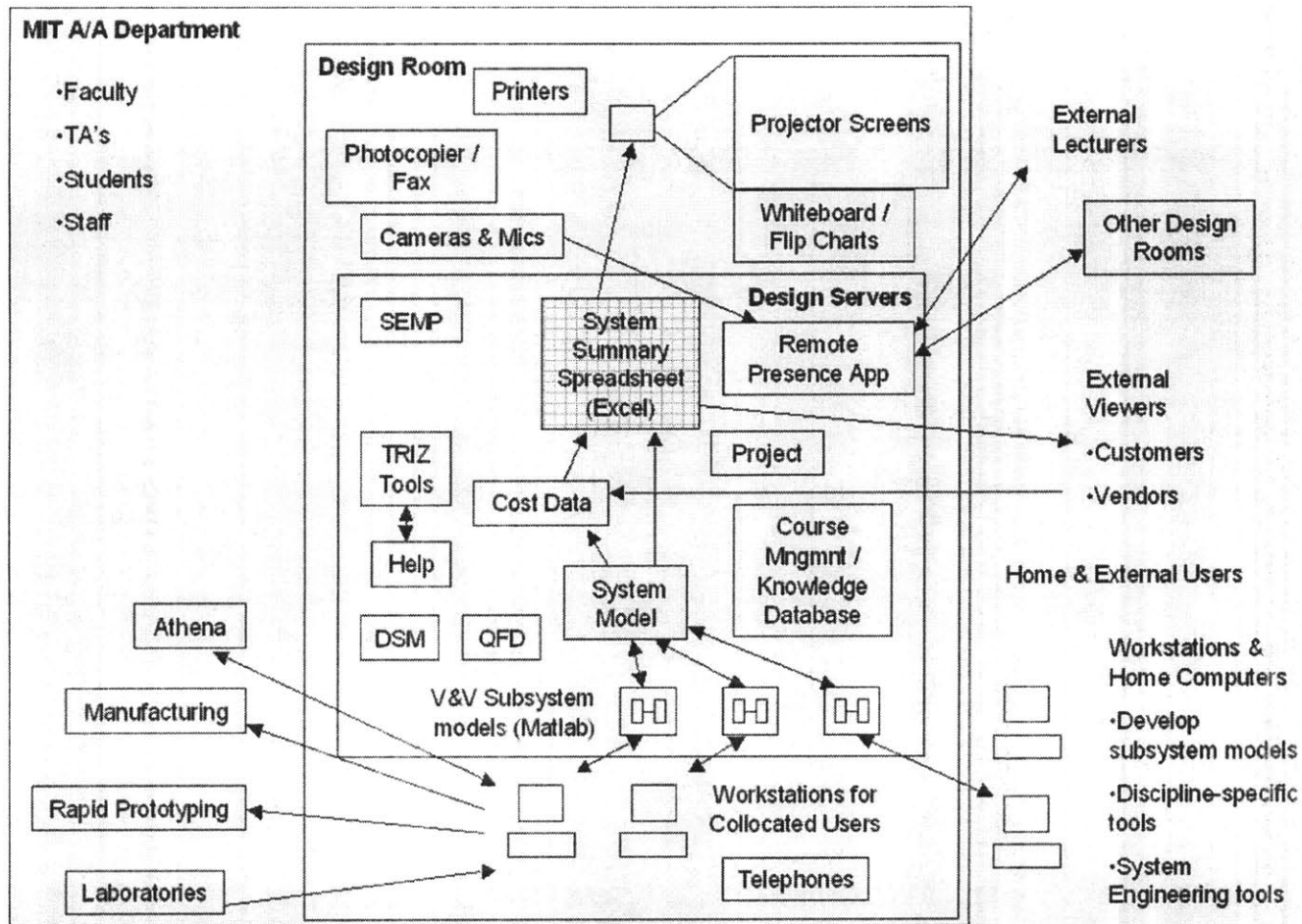
F.2 Architecture A Software/IT Conceptual View (see Farnworth [17])



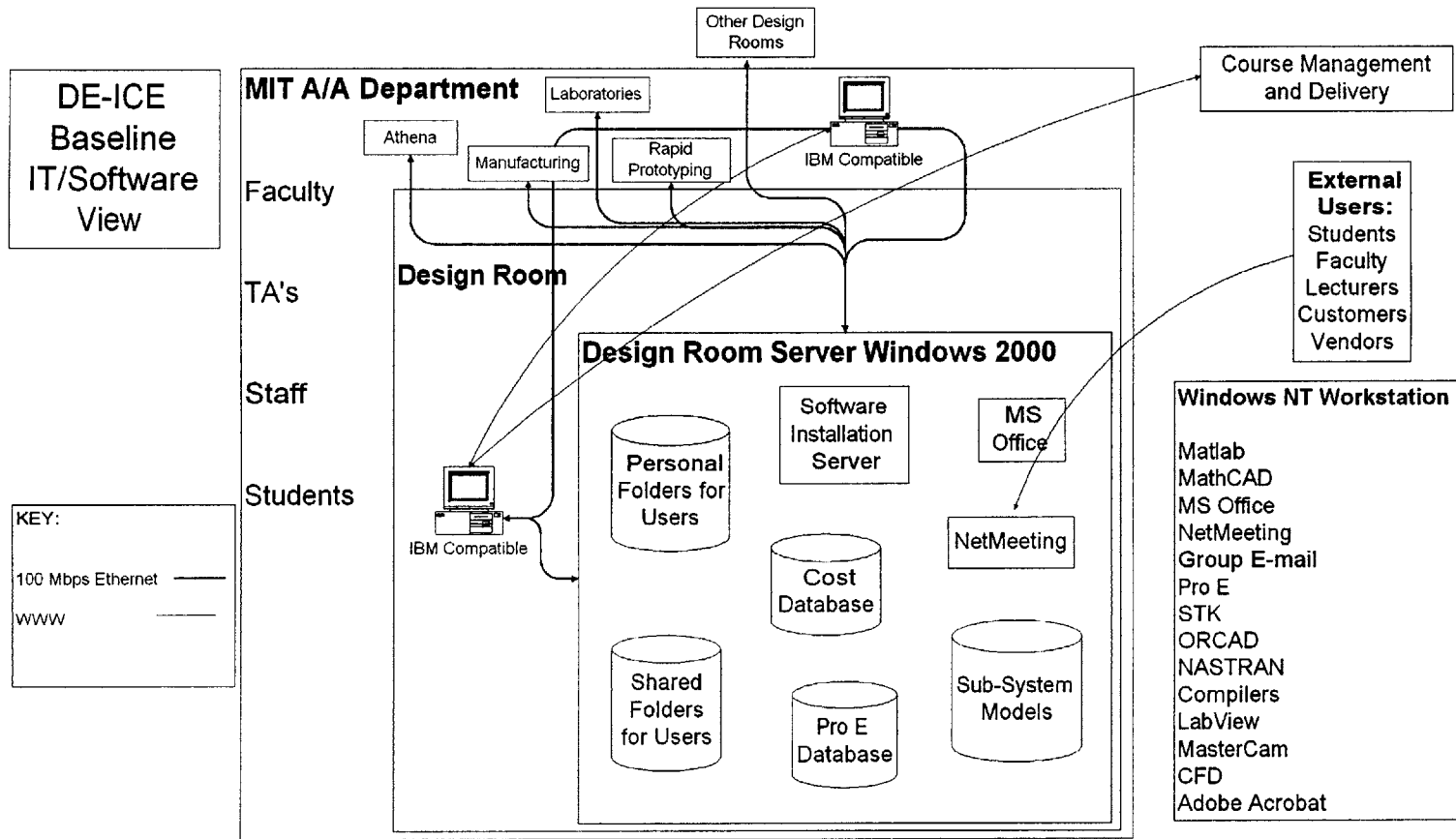
F.3 Architecture A Equipment Block Diagram (see Farnworth [17])



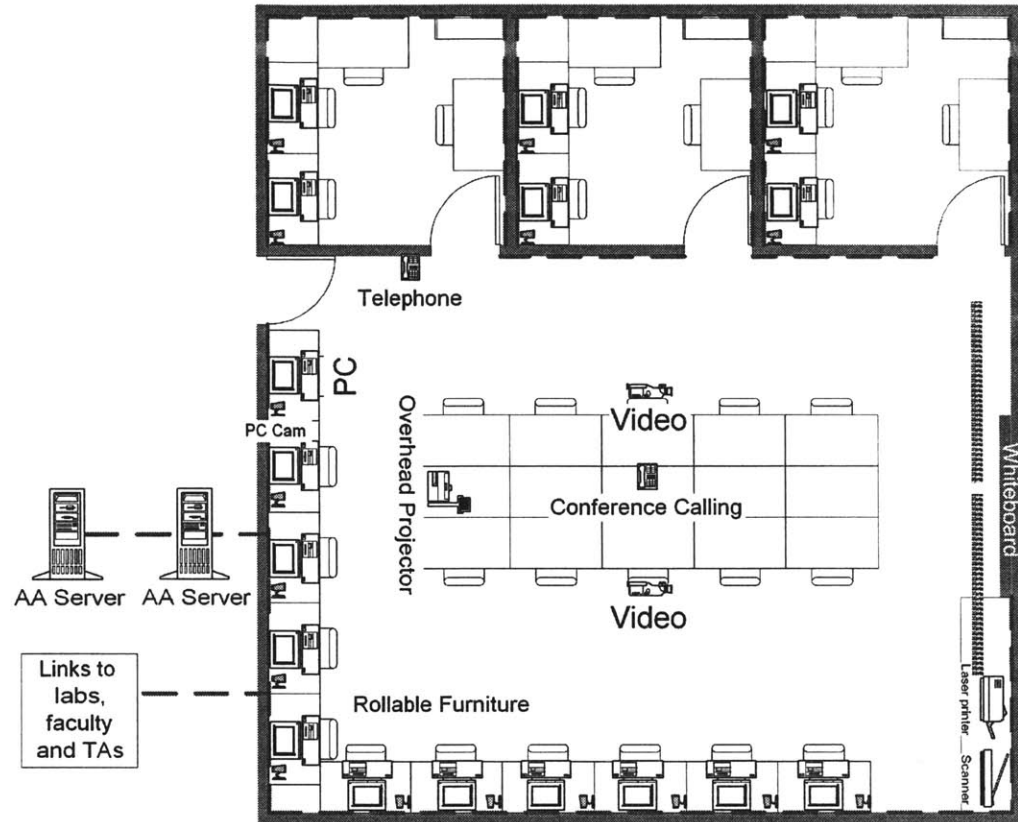
F.4 Architecture C Software/IT Conceptual View (see Manka [22])



F.5 Baseline Architecture Software/IT Conceptual View (see Farnworth [17])



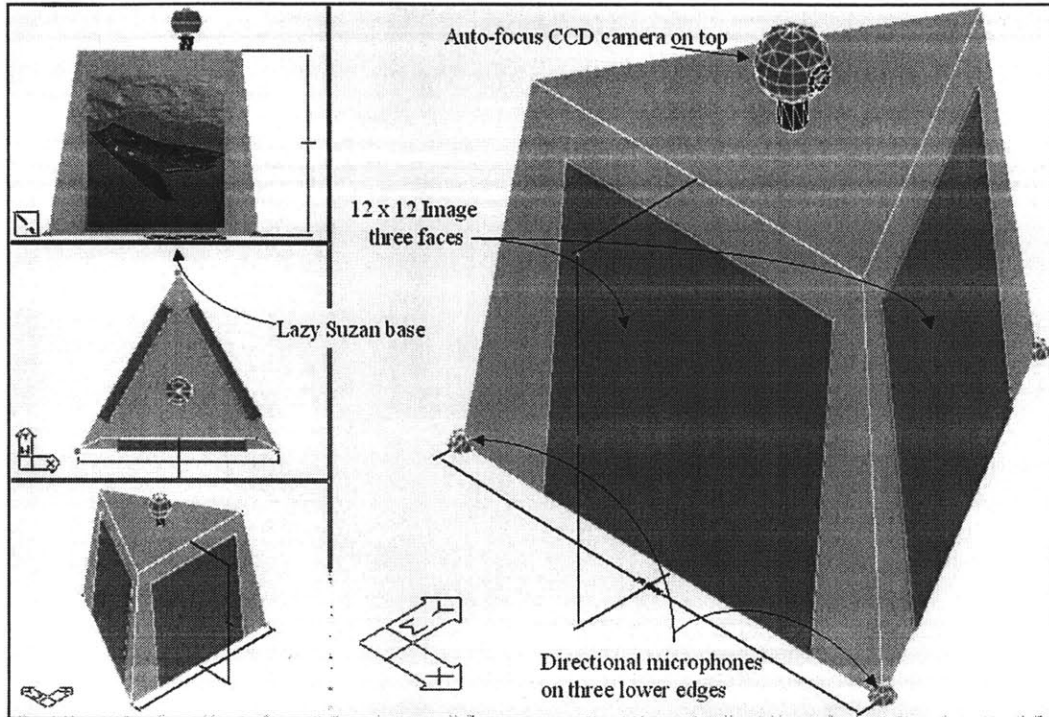
F.6 Baseline Architecture Physical View (see Farnworth [17])



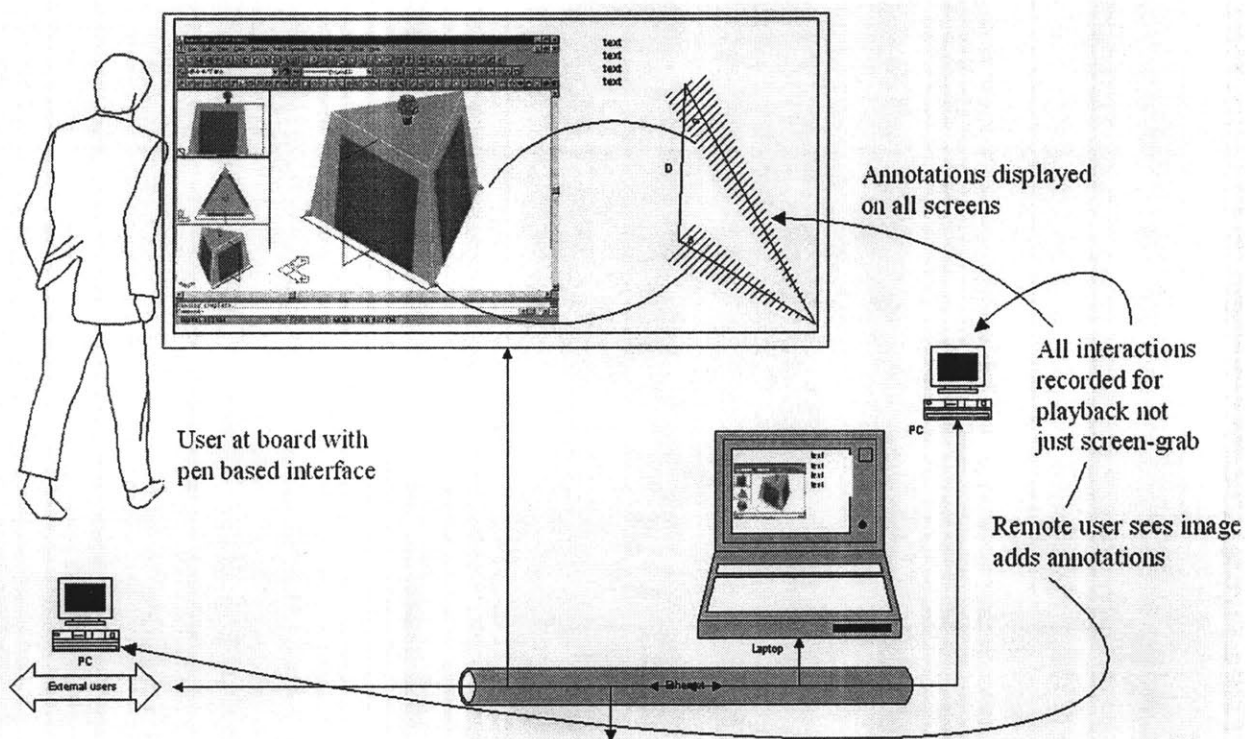
Appendix G

Prototyping Ideas

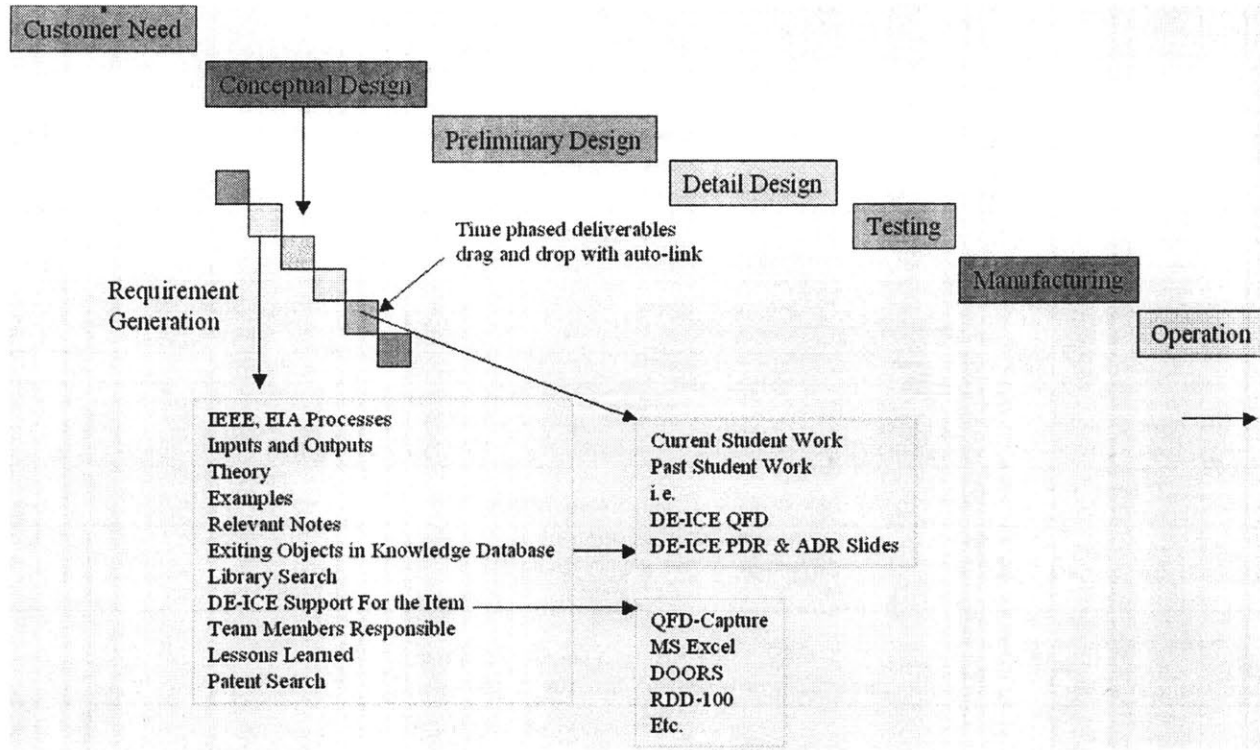
G.1 The Round Table Virtual Presence System (see Farnworth [17])



G.2 The Electronic Writeable Whiteboard (see Farnworth [17])



G.3 The Integrated On-Line TA (see Farnworth [17])



G.4 Prototype Breakdown (On-Line TA)

Team member	Priority	Task	Functionality	Demonstration of performance
Bruce	1	OLTA authoring tool (OLTA-AT)	Database with form to input PDP and content that will drive OLTA web-pages, define links to faculty and TAs, assign class deliverables, input class specific tools	Design display forms, database categories, database design, Show input and web-build
	2	OLTA front end	User interface driven by OLTA-TA, integrating all functions described in section 16.1 items j-x	Hardwired web-pages to show concept, modifiable content on a few pages from TA
	3	Build your PDP	Display tasks as hierarchical list indexed under top-level PDP, display task inputs and outputs, enable task selection by user, link to DSM with inputs from user selection	Display PDP database w/inputs outputs, selection of tasks, input to DSM
Alex	1	Object capture, knowledge database	Object tagging, object capture, knowledge database structure	Tags, file structure, student requirements, FFD
	2	Search OLTA	Integration of advanced search capability seeded from OLTA current task, search criteria user modifiable, capability to define search objects	Search prompt, FFD, Example search results
	3	Design tools	Testing evaluation and recommendations for design tool integration	Undefined
Simon	1	Submit class deliverables	Submission of class deliverables, interface to object tagging, input to knowledge database, possible auto-link?	Use case showing functionality, FFD
	2	DE-ICE support	Filtered list of tools available to support tasks at current point in PDP	Show tools supported at different level of the process, FFD
	3	Implementation plan	Simple short and long term plan to implement the DE-ICE system, focus on long term development of OLTA	Document showing short and long term plan as previously discussed

Appendix H

Demonstration

H.1 Sequence of Actions

Step	Location	Action 1 (click)	Action 2 (click)
1	work_manager.htm	pugh.doc	
2	attributes_open.htm	OK	
3	pugh.doc	back (on top)	
4	attributes_open.htm	back	
5	work_manager.htm	new	
6	work_manager_new.htm	PowerPoint	
7	attributes_new.htm	OK	
8	CDR.ppt	back (on top)	
9	attributes_new.htm	back	
10	work_manager_after_new.htm	3.0 Detailed Design	
11	detailed_design.htm	back (on top)	
12	work_manager_after_new.htm	CDR.ppt box	BringHome
13	attributes_bring_home.htm	back	
14	work_manager_after_new.htm	import	
15	attributes_import.htm	back	
16	work_manager_after_import.htm	Unlock File	
17	attributes_unlock.htm	OK	
18	work_manager_after_unlock.htm	CDR.ppt box	Update
19	attributes_update.htm	OK	
20	work_manager_after_update.htm	CDR.ppt box	Submit
21	attributes_submit.htm	OK	
22	work_manager_after_submit.htm	CDR.ppt box	Rename
23	attributes_rename.htm	back	
24	work_manager_after_submit.htm		

H.2 Work Manager Main Interface

Work Manager

Project	Project Element	Filename	Tool	Section to PDP	Last Modified	Lastest Version	Latest Submitted	Selection	Locked	
MINERVA	Trajectory Calculation					05/02/00	v1.2.2.3*	none	<input type="checkbox"/>	
		launch.m	Matlab	7.4.2 Matlab	05/02/00	v2.1.1*	none	<input type="checkbox"/>		
		orbit.m	Matlab	7.4.2 Matlab	04/11/00	v1.0	v1.0	<input type="checkbox"/>	A.M.	
		results.mat	Matlab	7.4.2 Matlab	03/25/00	v2.1	v2.1	<input checked="" type="checkbox"/>		
	Conceptual Design					04/19/00	v1.12.0*	v1.11.1	<input type="checkbox"/>	
		N2_EMCP.dmd	DeMaid	3.2.2 N2 diagrams	04/19/00	v3.0*	v3.0	<input type="checkbox"/>	B.F.	
		QFD_LTA2.xls	Excel	3.1.1 QFD matrices	03/25/00	v5.3	v2.4	<input type="checkbox"/>		
		launchcover.dwg	ProE	7.2.1 Technical drawing	02/29/00	v4.1*	v4.0	<input type="checkbox"/>	S.N.	

Buttons: New, BringHome, Import, Unlock File, Update, Submit, Rename, ProjNav, Close

Annotations:

- Open the file (points to QFD_LTA2.xls)
- Link to Tool (points to DeMaid)
- Link to PDP (points to 3.2.2 N2 diagrams)
- Means that modifications have been made since this version has been updated (points to v4.1*)
- Only one box is checked at a time (used to select file to update or submit) (points to checked box)

Open Version

Open Latest Modified

Open Version

4.1 [OK] [Cancel]

Update Version

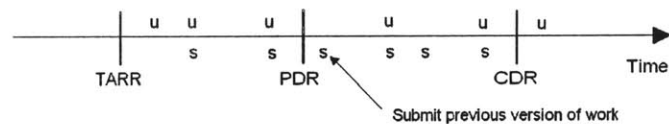
Enter New Version Number

4.2 [OK] [Cancel]

Submit Work

Enter Version Number to Submit

4.1 [OK] [Cancel]



H.3 Function “7.6 Open Current Work” Interface

Open Current Work

Filename / Project Element / Project:

Open Latest Modified

Open Version

Open for Editing

Open Read-Only

H.4 Function “7.5 Start New Work” Interface

Start New Work

PowerPoint document

First Name:

Last Name:

Project:

Project Element: Current: New:

Filename:

Type of Deliverables: Current: New:

Accessibility:

For the Project Navigator, enter input and output files, multiple files are allowed (optional):

Input Files

Output Files

H.5 Function “7.13 Bring Work Home” Interface

Bring Work Home

Filename / Project Element / Project:

Bring Latest Modified

Bring Version

Select one of the following:

Bring a Copy Only

Bring a Copy and Lock the File / Element / Project

Destination:

Email Address:

Zip Disk

Floppy Disk

H.6 Function “7.11 Import External Work” Interface

Import Work

First Name:

Last Name:

Project:

File Location:

File Destination:

Project Element: Current: New:

Updating an Existing File:

Creating a New Work Instance:

Type of Deliverables: Current: New:

Accessibility:

For the Project Navigator, enter input and output files, multiple files are allowed (optional).

Input Files

Output Files

H.7 Function “7.14 Unlock File” Interface

Unlock File

Filename / Project Element / Project:

Enter Modifications on File:

H.8 Function “7.7 Update Version in Database” Interface

Update File

CDR.ppt

Update to version:

H.9 Function “7.8 Submit an Assignment” Interface

Submit File

CDR.ppt

Submit version: v1.0

<Back OK

H.10 Function “7.16 Rename an Entity” Interface

Rename Work

Filename / Project Element / Project: CDR.ppt

Rename to: CDR.ppt

Enter New Accessibility (optional): Personal

<Back OK

Appendix I

Integrated Information Systems Work Proposal [53]



integrated information systems
innovations for the new economy™

MIT-Aerospace OLTA (On line Teaching Assistant)



PREPARED BY:

Karen Rankin
Enterprise Solutions Executive
Integrated Information Systems
25 Corporate Drive
Suite 202
Burlington, MA 01803
781-743-2014

www.iisweb.com

©COPYRIGHT 2000 Integrated Information Systems, Inc. All rights reserved.
This material may not be copied or reproduced by any means except for the express purpose of preserving back-up copies or creating additional work copies for internal use only. Unauthorized distribution of this material is strictly prohibited.

May 15, 2000

Mr. Simon Nolet
MIT
77 Vassar St.
Cambridge, MA 02142

Dear Simon:

The MIT Aerospace Engineering Group, as part of its continuing effort to provide the best possible education experience and engineer the best products possible, would like to enhance the way they deliver their course and product materials. As a means to achieve this, they would like to implement an On-Line TA solution. This solution will enable them communicate and share knowledge more efficiently between faculty, students and external resources.

Integrated Information Systems, utilizing our proprietary service methodology, can conceptualize, design, implement and support the right solution with the speed to market necessary to succeed in the rapidly evolving Internet environment for the MIT Aerospace Engineering Group.

An innovative e-commerce consulting firm, *iis* is leading the industry in custom software solutions and network integration. This is because our seasoned professionals—with a large number of certifications, a deep understanding of strategic business issues, and a focus on training for new and evolving technologies—are committed to personalized service. We establish long-term relationships by developing an intimate knowledge of our clients' businesses, enabling us to leverage their internal technical resources as part of a successful conclusion. Our goal is to go beyond the traditional client/vendor relationship by striving to exceed expectations through a proactive approach to project management.

iis appreciates the opportunity to present this proposal to the MIT Aerospace Engineering group. We are confident that our comprehensive solution and management approach will satisfy your requirements, and we look forward to working with you on this project.

Very truly yours,

Karen Rankin
Enterprise Solutions Executive

Table of Contents

1.0	Proposal Summary	5
1.1	STAFF LOCATION.....	6
1.2	COMPANY PROFILE.....	6
1.3	<i>iis</i> QUALIFICATIONS.....	6
1.3.1	<i>The iis Approach</i>	6
1.3.2	<i>The iis Difference</i>	6
1.3.3	<i>Company Recognition</i>	7
1.4	PROJECT TEAM.....	7
1.5	PROJECT MANAGEMENT PROCESS	7
2.0	Project costs and timing	10
2.1	ESTIMATED COSTS.....	10
Appendix A: Examples of iis work		11
Appendix B: Strategic Partnerships		13
Appendix C: Manufacturer Certifications		15

1.0 PROPOSAL SUMMARY

MIT Aerospace Engineering Group has approached *iis* looking to establish a partnership for the development of an enhanced On Line TA system to manage the Product Development Process for MIT's aerospace curriculum.

iis is a full-service Internet professional services provider focused exclusively on the digital transformation of businesses. We provide advanced Web-based solutions that distribute the power of information. The common focus of our technology solutions revolves around the Web. Because we understand the value of information, our consulting services cover all aspects of information management including strategy, application development, network architecture, and business intelligence.

The following information was provided by MIT Aerospace Engineering Group and generally describes the purpose and requirements of this project.

Purpose:

This project's initiative is to improve communication, promote knowledge transfer, and facilitate a more efficient Project Development Process throughout the aerospace engineering community. As this objective is achieved the aerospace engineering students will then be better able to produce results consistent with the mission of the Massachusetts Institute of Technology. "To advance knowledge and educate students in science, technology, and other areas of scholarship that will best serve the nation and the world in the 21st century."

General Solution Requirements:

- Fast and easy system for faculty to create and customize class specific content
- Web based user interface (OS independent)
- Consolidate information for patent searches, knowledge base and Product Development Process
- Allow full access to all resources on the DE-ICE system
- Secure manageable access control
- Reduce the time requirement on faculty to 1/10 of normal class preparation time
- Allow students more flexibility to manage their work

Timeline:

- August 2000 – Pilot for Fall Semester

Based on our proprietary development model, we address client needs in a manner that ensures a common vision for the project's completion. *iis* will begin the MIT Aerospace Engineering Group e-business project at the Explore Phase to better define the technical requirements to develop an information architecture. The Explore Phase enables *iis* to better understand the breadth of the project and therefore document specific requirements critical to the project's success. *iis* will deliver back to MIT Aerospace Engineering Group the following documents, which in turn will be used to properly define the scope, functionality and price the MIT Aerospace Engineering Group project:

The **Explore Phase** is designed to discover and detail information pertinent to the following areas:

- Existing Information architecture – data sources, access methods, schemas and process flows
- Platform requirements – system capabilities, requirements and scalability
- Functionality requirements – security & access control, usability & flexibility, pricing models & options, search criteria & capabilities and information presentation.

Deliverables:

- Functional Requirements Document – serves as the application's the functional roadmap
- Project Plan – provides measurable timelines, milestones and defines resources
- Detailed Proposal – details project costs

iis will accomplish each of the outlined objects by employing our proven Project Methodology defined in Section 3.9. Our methodology begins with the Conceive Phase. In this case, we feel this phase can be bypassed due to the outstanding thought work, research and documentation effort that has been already been conducted by Simon Nolet, Bruce Farnworth and Alex Manka.

Based on the foundation built in the Explore phase, the OLTA site will be designed, built and launched. The work done in the Explore phase will result in the ability to accurately assess the costs for development, quality assurance and launch of the site. The timeframe and cost for the Explore phase has been estimated in Section 3.0, Project Costs & Timing. A ballpark estimate of cost for the balance of the project has also been provided. A detailed estimate of cost for the balance of the project will be provided at the end of the Explore phase, based on the level of detail uncovered in that phase.

1.1 Staff Location

iis currently has 550 employees. Company headquarters are located in the Tempe, Arizona, and we have regional offices in Atlanta, Denver, Las Vegas, Los Angeles, and Boston.

1.2 Company Profile

Since *iis* was founded in 1988, we have put together a staff of highly experienced, certified professionals who are not only technical experts, but who also have business savvy with years of industry experience and advanced business degrees. This includes an extraordinarily high percentage of MBAs, plus business veterans who have spent ten or more years succeeding in corporate environments.

iis believes its size, pace of growth, and strong reputation in the technology community position the company as a leader in the information technology marketplace. *iis* was an early implementer of Client/Server technologies and an early leader in Internet, Intranet, Extranet, and Electronic Commerce solutions beginning in 1997.

1.3 *iis* Qualifications

1.3.1 The *iis* Approach

Our business model has two major approaches that place *iis* in a unique category of IT service providers: our approach to services, which combines business expertise and technological innovation, and our proprietary *iis* DimensionsSM methodology, which features rapid development and deployment.

The *iis* business style involves the client at every stage of the development process. At the foundation stage, we work with your team to determine what business results you expect from the solution. Building on this foundation, we work together to further define a vision for the prototype application. Using an approved prototype, we then engineer your solution. When the system goes live, our project team is with you to ensure a smooth transition.

1.3.2 The *iis* Difference

What serve to distinguish *iis* from today's growing crowd of IT consultants are these key qualities: our philosophy, our people, and our proprietary methodology.

iis is continually evaluating the services offered in light of the needs of our current customers and the needs of an ever-changing marketplace. Currently, there are five key areas in which we provide services—Strategy Consulting, Customer Experience Design Solutions, Application Development and Integration, Internet Infrastructure Services, and Application Management Services.

1.3.3 Company Recognition

Recently, *iis* ranked 457th on the prestigious Deloitte & Touche Technology *Fast 500* list of the fastest growing technology companies in the U.S. *iis* was also included in the 1997 and 1998 *Inc. 500* list of the fastest growing private companies in America. Microsoft has recognized *iis* as its *Microsoft Southwest District Solution Provider Partner of the Year* for 1998 and 1999. In 1999 the Arizona Software Association named Jim Garvey, *iis* CEO, their *CEO of the Year*.

1.4 Project Team

We anticipate that the *iis* Project Team for the Explore Phase would consist of the following functional roles:

Solution Architect

The Solution Architect has both strong technical capabilities and communications skills in order to translate a client's complex business problems into innovative technical solutions. As both a leader and role model, the SA provides technical leadership on projects from beginning to end, with emphasis not only on excellent initial design but also on creative and efficient problem resolution.

Creative Designer

iis Customer Experience Group (CEG) Senior Designers assist our clients in creating an effective online presence rooted in the best interest of their end users. Information Architecture, Page Layout, Creativity, and a User-Centered design approach are prerequisites to be a part of the CEG team. Through co-production, innovation, and interactive design sessions, Senior CEG designers work in concert with an initiative's project team and clients to produce a visual design which meets the needs of the initiative and exceeds the expectations of its users.

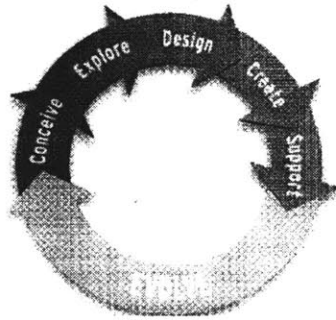
Project Manager

The Project Manager is responsible for developing strong relationships with clients; leading, motivating and focusing their team; understand how to apply technology to solve the client's business problems.

Specific team members will be assigned when the project is awarded to *iis* and an exact timeframe is established.

1.5 Project Management Process

iis will take a phased approach to developing the *she-co.com* website. A phased approach keeps the project manageable and the project team focused, and ensures that the design and development of systems/solutions are based on clearly defined business objectives and requirements. Though individual projects vary substantially according to client needs, the description outlined below illustrates the usual process, the potential tasks, and the potential deliverables for each phase of the *iis* approach.



Phase 1: Conceive

We will work together to develop your new web site strategy from the ground up with an *iis* Next Dimension Innovation Workshop. Working in an atmosphere of imaginative thinking and open dialogue, we will size up your critical issues and explore how to address them. Through this collaborative, idea-engaging process, we will define a website vision for *she-eo.com* and develop a comprehensive plan for implementation.

The workshop will define the project scope while focusing on such issues as market and technology trends, competition, and future business conditions. Also included in the process are any legacy system issues, your financial requirements, and your required timelines. The emphasis is on providing clarity prior to the solution development. In addition to refining your Web strategy prior to project initiation, this Innovation Workshop jump-starts the development of your e-business site. Our Project Team will work with your Customer Experience partner to generate ideas and demonstrate all the possibilities available. This workshop is especially beneficial because it gives *iis* the opportunity to share its Internet expertise with your organization and provide insight into how you can best utilize the Web as an e-business tool.

Phase 2: Explore

The Explore phase defines the parameters for the engagement's success by establishing the preliminary project scope, business and functional requirements, and preliminary project plans. Tasks include defining project scope, business requirements, functional requirements, and project metrics (i.e., cost, schedule, etc.). During this phase, *iis* will deliver a Preliminary Project Scope Document, A Business Requirements Document, a Functional Requirements Document, and a Preliminary Project Plan.

Phase 3: Design

The Design Phase defines, based on the business and functional requirements, the architecture, algorithms, data models, object models, use cases, visual interfaces, and test plans for the projected system. *iis* professionals define user interfaces, algorithms, data models, object models, use cases, visual interfaces and the test plan, and refine project planning. During this phase, the team delivers items such as a Customer Interface Document, a Technical Design Document, a Test Plan, and a Project Plan.

Phase 4: Create

During the Create Phase, *iis* constructs the client's system based upon the *iis* design documentation and provides testing based upon client-approved system requirements. The *iis* team will set up the system's infrastructure, develop the code, develop production graphics, and create the database. In this phase, they will also carry out unit testing, system integration testing, design testing, (i.e., load and performance testing), data conversion, and requirements testing. Deliverables may include a Finished System, a Design Test Report, and a Requirements Test Report.

Phase 5: Support

The Support Phase is defined by the delivery and support of the client's system in line with the original *iis*-developed e-business strategy, and involves the tasks of implementation, training, maintenance, hosting/management, and enhancement. The system is delivered, often along with a Training Plan and a Maintenance and Support Plan.

Phase 6: Evolve

The Evolve Phase provides closure for all project development activities for the release and proposes, simultaneously, direction for the next potential phases of the solution. This is accomplished through a review of the solution based on the previously defined success criteria, review of the business requirements, customer surveys/feedback, a project configuration closure, and a Client-involved "lessons learned" session. Deliverable documents may include a Survey Report, a Review and Analysis Report, a Final Project Report, and a project archive.

2.0 PROJECT COSTS AND TIMING

2.1 Estimated Costs

iis believes in taking a holistic approach to solving client's e-business needs. We accomplish this by ensuring that business design and strategy is addressed at the initial stages of a client engagement. Our e-Business Architects will help MIT Aerospace Engineering Group define the business requirements and value proposition of their Web initiative based on the experiences to be delivered to MIT Aerospace Engineering Group users. The MIT Aerospace Engineering Group value proposition will then dictate the technical and creative development of the Web application.

The Explore Phase will be executed by a Solutions Architect, Creative Designer and Project Manager and the cost estimate is provided below. We expect that this phase can be completed in a three-week timeframe consisting of a half-day meeting, independent working sessions, review of draft documentation, editing and follow up sessions and presentation of the materials.

In addition, an estimate for the implementation of the solution has also been provided. It is anticipated that services for the complete project's implementation would be provided by a Solutions Architect, Sr. Consultant, Designer, Design Implementer, Project Manager and Database Administrator, but a final staffing plan would be determined at the end of the Explore phase. Although we have provided this estimate as a guideline, *iis* is willing to discuss options to work the OLTA project team to determine the most effective options for utilizing MIT and *iis* resources.

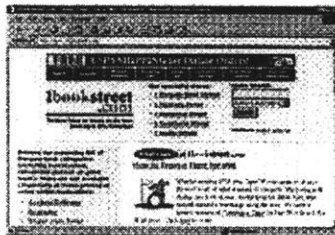
Phase	Estimated Time Frame	\$
Explore Phase Estimate*	3 weeks	\$15,500
Project Implementation Estimate* (including the Design & Create phases)	12 weeks	\$306,000
*Work to be done on Time & Materials Basis	Total	\$321,500

This estimate is based on the information provided to *iis* to date. Following the Explore Phase, a more accurate estimate for development and launch of the OLTA site will be provided. Production environment hardware and software costs are not included in the estimate.

APPENDIX A: EXAMPLES OF IIS WORK

VISITALK.COM

iis provided project management, analysis, requirements definition, design, development, testing, and implementation for the visitalk Internet Web site. The goal of the site is to connect people around the world to share voice, video, and information via the Internet. This solution allows members to record, play, and send voice messages over the Internet to anyone listed in its global directory.

**1BOOKSTREET.COM**

iis provided project management, analysis, requirements definition, design, development, testing, and implementation for the Soda Creek Press Internet Web site. Since launching www.1bookstreet.com in September of 1998, Soda Creek Press has expanded its customer base by over 200% and their Internet revenues have increased twentyfold.

ORMESCHOOL.ORG

iis provided project management, analysis, requirements definition, design, development, testing, and implementation for the Orme School Internet Web site. The Orme School was pleased with *iis* technical expertise and performance on a networking project and felt comfortable addressing the Web site challenge with us as well. Equally important was the fact that the school was able to leverage promotional dollars they had previously spent.





iis provided project management, analysis, requirements definition, design, development, testing, and implementation for the CSK Auto Internet Web site. The solution provided an easy-to-use self-service customer order channel to purchase automotive parts on-line from a catalog of over 1.5 million parts.



iis provided project management, analysis, requirements definition, design, development, testing, and implementation for the artup.com Internet Web site. The solution included an auction site for artup.com clients, a business-to-business channel for members, a private pricing structure for artists, and the ability artists to upload works from an Internet browser. The site also features an approval process for artup.co verify members and artwork as well as the ability to process all credit card sales and handle accounting functions.

APPENDIX B: STRATEGIC PARTNERSHIPS

ii' business focus is **vendor-neutral**. We provide hardware and/or software only when our customers specifically ask us to do so. However, in order to maintain our technical excellence and strong support relationships with the technology vendors we integrate, we maintain strategic partnerships with the following companies:

IBM (Premier BESTeam Member): *ii* has been partnering with IBM for the past three years. IBM BESTeam members are recognized as experts in delivering IBM-based solutions in Operating Systems, Application Development, Data Management, Work Management, Networking, Internet, Systems Management, and Transaction Systems. As a Premier BESTeam Member, *ii* enjoys such benefits as: direct communication to top-level IBM technical support not available to the general public; advanced copies of software and code to enable rapid adoption and deployment of IBM technology solutions; and access to unlimited private training in IBM products, taught by IBM inside experts. *ii* has 17 consultants trained and certified on DB2, 6 consultants certified on Net.Commerce, 12 people trained in the Websphere tools, 6 people trained in Visual Warehouse, and 2 CSEs.

Microsoft Certified Solutions Provider Partner ("MCSP"): *ii* has been partnering with Microsoft for six years. *ii* has proven Microsoft product expertise and experience with a multitude of Microsoft products and technologies. As an MCSP, *ii* enjoys the highest level of engagement with Microsoft through a variety of exclusive benefits like business development tools, early access to Microsoft products, technical information and resources, and worldwide networking with other MCSP organizations. *ii* has 25 MCSEs, 30 MCSDs, and 10 MCTs.

Sybase: *ii* has been in partnership with Sybase for five years. *ii* is authorized to sell, install, and configure Sybase database products as well as connectivity and application development tools. *ii* has 6 Sybase DBAs and 40 consultants experienced with Sybase database products.

Novell (Platinum Partner): *ii* has been partnering with Novell for seven years. *ii*' consultants hold numerous certifications for the different Novell products available. Our partnership allows us to resell, install, and configure these products.

Oracle (Business Alliance Partner): *ii* has been partnering with Oracle for five years. *ii* is authorized to sell, install, and configure Oracle database products as well as connectivity and application development tools. *ii* has 6 Oracle DBAs and approximately 50 consultants that have been through various levels of Oracle training and have experience with Oracle database products.

EMC Corporation: *ii* has a strategic relationship with EMC that involves partnering on project opportunities to bring total project solutions to EMC customers involving e-commerce, business intelligence, and client/server initiatives, providing training to EMC clients, and engaging in joint presentations and seminars.

FileNET (Gold Authorized Reseller): *ii* has been partnering with FileNET for three years to implement, customize, and configure FileNET's integrated suite of client/server products for enterprise document management.

Informix: *ii* has been partnering with Informix for four years. *ii* is authorized to implement, customize, and configure Informix database products, connectivity, and application development tools.

Lotus: *ii* has been partnering with Lotus for six years. *ii* holds numerous certifications for the Lotus Notes products. *ii* can implement, customize, and configure the many Notes product lines.

Sun Microsystems: *iis* is in discussions with Sun Microsystems regarding becoming a Sun Authorized Java Center. The services offered at an Authorized Java Center bring together proven and successful Java expertise and can guide customers toward the best Java-based implementations and solutions. In addition, *iis* has many experts trained in Networking, Web/Internet Development on the Sun Solaris platform and has worked on more than 30 projects using Sun Solaris 2.6. *iis* is a member of the Sun Developer Connection as well as the Java Developer Connection. *iis* has developed an extensive four-module program for Java certification and training of its consultants.

Baan: *iis* has been partnering with Baan for one year. *iis* is an authorized implementer and reseller of Baan products.

Sagent: In the Business Intelligence area, *iis* has been partnering with Sagent for one year to implement, configure, and customize its products.

Cognos: In the Business Intelligence area, *iis* has been partnering with Cognos for one year to implement, configure, and customize its products.

Sales Logix: In the customer relationship management area, *iis* has been partnering with Sales Logix for one year to implement, configure, and customize its products.

Siebel: *iis* has been partnering with Siebel for one year to implement, configure, and customize its customer relationship management software products.

Onyx: *iis* has been in partnership with Onyx for two years to implement, configure, and customize, its customer relationship management software products.

HP Openview: *iis* has been partnering with HP Openview for one year to implement, customize, and configure its products.

Wyse: *iis* has been in partnership with Wyse for two years to implement, customize, and configure its products.

Citrix: *iis* has been partnering with Citrix for one year to implement, customize, and configure its products.

Cisco: *iis* has been partnering with Cisco for one year to implement, customize, and configure its products.

iis also has numerous strategic relationships and business partnership with niche software and service vendors including the following:

- Ariba (business to business e-commerce procurement);
- Cyclone Software (an XML trading engine for business to business e-commerce solutions and supply chain management);
- e-docs (the market leader in electronic bill payment and presentment);
- Broadvision (a market leader in e-commerce solutions);
- Usinternetworking (the nation's largest application service provider);
- UPS Logistics (the leading organization for logistics management);
- CommercialWare (the market leader in e-commerce solutions for the catalog and direct sales industry);
- and
- Qwest/US West (a strategic partnership to provide e-commerce related services to Qwest/US West clients).

APPENDIX C: MANUFACTURER CERTIFICATIONS

Company-wide, we maintain certifications in the following technologies:

- Databases
 - Certified Oracle DBAs
 - Certified Sybase DBAs)

- Programming Languages
 - Certified PowerBuilder Developers
 - Lotus Notes Certified—Consultant, Developer, and Specialist
 - Microsoft Certified Solution Developers
 - Borland Delphi Certified Developers

- Network Engineering
 - Microsoft Certified Systems Engineers
 - Certified Netware Engineers

- Trainer Certifications
 - Borland Delphi Certified Trainer
 - Microsoft Certified Trainers

- Development Platform
 - IBM CSE Net.commerce Certified
- Additional Certifications
 - Baan
 - Baan Finance—Advance
 - Certified Netware Administrators
 - Certified Production Inventory Management
 - Certified WinFrame Administrator
 - Cognos
 - Cognos Certified Impromptu Administrator
 - Cognos Certified Powerplay Administrator
 - Cognos Certified Impromptu Professional
 - Compaq Professional Sales Certification
 - Configuration Management: CMII Certification
 - FileNET Certified Professional
 - IBM Visual Warehouse 5.2
 - Integrated Resource Management
 - JADE (1)
 - Microsoft Certified Database Administrator
 - Microsoft Certified Product Specialists
 - Microsoft Certified Professionals
 - Microsoft Certified Professional + Internet
 - Project Management Professionals
 - Rapid Application Development Certification
 - Solutions Expert DB2 UDB V5 Database Administrator
 - SPHR

2700-19