

1)

# Evaluation of Lotus Notes and Autocad for Collaborative Design

by

**Gary Cheng**

B.S. Civil Engineering  
University of California, Berkeley  
1995

Submitted to the Department of Civil and Environmental Engineering  
in Partial Fulfillment of the Requirements for the Degree of

**MASTER OF ENGINEERING**  
in Civil and Environmental Engineering

at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**  
June 1996

© 1996 Massachusetts Institute of Technology. All rights reserved.

Signature of Author \_\_\_\_\_  
Department of Civil Engineering  
May 10, 1996

Certified by \_\_\_\_\_  
John R. Williams  
Professor of Civil and Environmental Engineering  
Thesis Supervisor

Accepted by \_\_\_\_\_  
Joseph M. Sussman  
Professor of Civil and Environmental Engineering  
Department Committee on Graduate Studies Chair

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

JUN 05 1996

ENG

LIBRARIES

# **Evaluation of Lotus Notes and Autocad for Collaborative Design**

Gary Cheng

Submitted to the Department of Civil and Environmental Engineering  
on May 10, 1996 in partial fulfillment of the requirements for the degree of  
Master of Engineering in Civil and Environmental Engineering

## **ABSTRACT**

There have been many studies on the issue of how an individual begins to design a building or object. But today, multidisciplinary design teams are usually the norm and it is therefore vitally important that they be able to communicate and share information in order to facilitate the collaborative design process. But the technological issue is not the only inherent problem that plague these teams. Each discipline brings with it its own knowledge tree of expertise; often times making it difficult for everyone to view and think about a problem in the same way.

The purpose of this work is to examine both a groupware and design tool and how each contributes individually and collectively in assisting the collaborative design process. Lotus Notes and Autocad were chosen because each is an industry leader as a groupware and design tool respectively. By integrating the collective attributes of each of these software systems, the results will be examined on how these tools can aid or even hinder a collaborative design process by a multidisciplinary team.

The last section will focus on an actual class at MIT that took place which was developed to study these issues. Teams consisting of both engineers (civil and mechanical) and architects were formed whose purpose were to not only use the tools as described above but also to pool their collective yet decentralized knowledge for a collaborative design project, starting from conception to a actual scale model.

Thesis Supervisor: Prof. John R. Williams  
Professor of Civil and Environmental Engineering

# Table of Contents

## CHAPTER 1 - OVERVIEW

1.1 Introduction .....	5
1.2 Civil Engineering and Collaboration .....	5
1.3 Thesis Focus .....	6

## CHAPTER 2 - INHERENT COLLABORATION PROBLEMS

2.1 Introduction .....	7
2.2 Individual Perspective .....	7
2.3 Abstraction .....	8

## CHAPTER 3 - GROUPWARE

3.1 Definition .....	11
3.2 Communication .....	11
3.3 Thinking .....	12
3.4 Information Availability .....	13
3.5 Time-Space Taxonomy .....	14
3.6 Software System .....	15

## CHAPTER 4 - LOTUS NOTES

4.1 Overview (Notes as a Strategic System) .....	17
4.2 Document Database .....	17
4.3 Replication .....	19
4.4 Security .....	21
4.5 Future .....	23

## CHAPTER 5 - AUTOCAD

5.1 Introduction .....	25
5.2 Abstraction .....	25
5.3 Graphical Representation .....	27
5.4 Object Database .....	29

## CHAPTER 6 - CASE STUDY

6.1 Introduction .....	32
6.2 Technology Use .....	33
6.3 Interaction .....	35

CONCLUSION .....	37
------------------	----

APPENDIX .....	39
----------------	----

REFERENCES .....	50
------------------	----

## List of Figures

FIGURE 1: KNOWLEDGE TREE.....	9
FIGURE 2: KNOWLEDGE OVERLAP .....	10
FIGURE 3: COMMUNICATION MODEL [REF. 3].....	12
FIGURE 4: INFORMATION MODEL.....	13
FIGURE 5: 2x2 TIME-SPACE MATRIX.....	14
FIGURE 6: NOTES DOCUMENT.....	19
FIGURE 7: CAD LEVEL ABSTRACTION .....	26
FIGURE 8: EXAMPLE OF IFC RELATIONSHIP .....	31

# **Chapter 1 - Overview**

## **1.1 Introduction**

Collaboration has been a key concept these last few years; with the widespread availability of technology and communication mediums, it is now possible for people to work collectively toward a common project from anywhere in the world. The potential of the group is always greater than the sum of its individual members - “that is the magic of a group, that is its real added value” [Ref. 1]. Historically, culture has shown that the user usually goes to the data; i.e. person has to go to the office to work because that is where the computer and database of information are located. A paradigm shift has taken place whereby that data follows the user; now that same person equipped with a personal laptop computer has the same information as before without even having to go to the office. The increased mobility and flexibility of people has added to the critical need of developing systems to allow teams to work collaboratively without being in the same office. However, the communication tools and software developed were aimed mostly at the business world like consultants or a big corporation which needed them. Unfortunately, it has been a difficult road for the civil engineering community to make inroads in this endeavor.

## **1.2 Civil engineering and collaboration**

Most of what is out in the marketplace in terms of collaboration tools for civil engineering is a potpourri of software applications. Many civil engineers have almost always had to develop their own engineering software tools because those in the market today just are not good enough or are not specialized enough for the particular project

that they are working on. As a consequence, there lacks any standards or framework to develop a cohesive, integrated set of civil engineering software tools. There are many tools based on old systems platforms like MS-DOS, which although are adequate as a standalone application, lack the ability to integrate with other more modern based operating systems like MS-Windows which support Object Linking and Embedding (OLE) [Appendix 1]. With this inherent problem, the civil engineering community has had to mix and match certain pieces of software packages and tools together, often times having to write code in order to accommodate the translation from one software output to another as the input.

### **1.3 Thesis Focus**

This thesis will focus first on the problems of cross-disciplinary collaboration between architects and engineers and the different planes of perception and understanding that each discipline has and how it affects the design process. Next, I will concentrate on two software applications which exemplify a groupware and engineering tool respectively, Lotus Notes and Autocad R13 and how they foster the collaborative design process. Finally, I will examine how teams, consisting of architectural and engineering students, in a virtual design studio class at the Massachusetts Institute of Technology are using these tools to help facilitate collaboration between themselves and to understand what issues may encourage or detract them from working together effectively.

## **Chapter 2 - Inherent Collaboration Problems**

### **2.1 Introduction**

Collaborative design is often a very difficult process when people from different disciplines or departments are put together for a project. Each discipline has developed its own unique culture and set of definitions on how to approach the same problem, each taking a different viewpoint. For example, an architect might focus on lighting or the “space” when examining a structure whereas an engineer would instead focus on the control or structural systems. Therefore, before looking at multidisciplinary collaborative design, one should examine the problems of design from an individual perspective and then this may hopefully provide us a framework to abstract the issues that affect collaborative design.

### **2.2 Individual Perspective**

What is the problem that hinders the design process, especially for individuals? Is it the sheer physical size of the project, the number of components to consider, or maybe the interactions between these components? All of these questions refer in some way to the physical properties of the project or design. The problem in design may not lay so much in the inherent properties of the “project”, but more in the “property of the person.” Therefore, the problem of design may depend on the state of knowledge of the designer; the problem is dependent on the designer’s view of the system.

If one is an expert in a specific system, then the design of it is not as complicated as compared to someone who may just have partial or passing knowledge of it. It is this

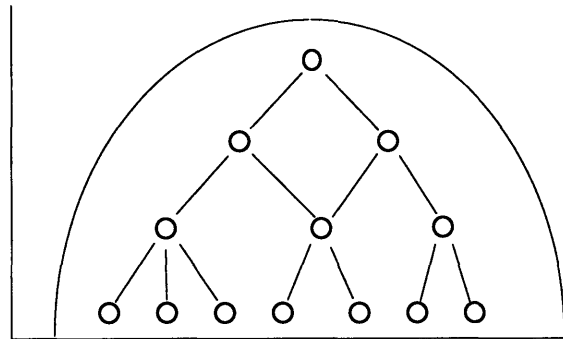
“state of knowledge” that enables a problem to become less complicated to a certain set of individuals who carry those specific skills to solve it. It has taken years of education and experience for each of these experts to gain this highly specific knowledge of a particular system. Now consider a design project requiring many multidisciplinary experts to solve the multitude of system problems that arise in a big project. Each expert will obviously be able to solve his/her own specific system problem, but the goal is to collaborate and somehow for everyone to communicate with each other and pool each other’s talent for the good of the whole team.

### **2.3 Abstraction**

Knowledge can be understood or viewed in a hierarchical tree, illustrated initially by thousands (or millions) of concrete details at the bottom; however the human mind cannot possibly keep track of all these details. Abstraction is the main way of overcoming this information overload complexity by collating and collecting the many details and generalizing it into a single idea or concept which is then easier to understand. This idea or concept encompasses all the properties of the objects from which it was abstracted. Below is an example which illustrates this point more clearly.

Consider the case of the animal taxonomy {hierarchy} where one can keep abstracting the animal kingdom up from a specific species on a low level towards a more general and broad naming level. Consider the example of humans and whales. Both of them can be abstracted to a higher level name called a mammal because each of them have something in common, they are both warm-blooded animals. Now one can say on a high level that all mammals are warm-blooded (qualities of humans and whales), but that on a

lower level, one quality that differentiates a human and a whale is that one lives on land and the other in the sea.



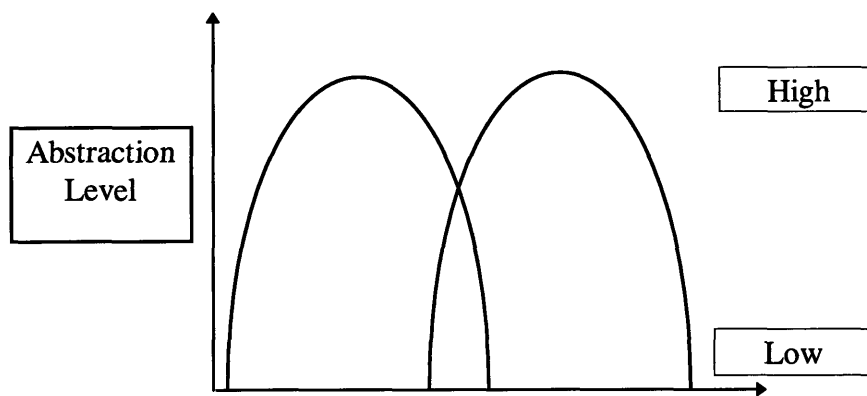
**Figure 1: Knowledge Tree**

As illustrated in Figures 1, the low level is represented by many details and these are eventually abstracted toward more generalized concepts. For example, each child node will inherit certain properties from its parent and will therefore become specialized at the bottom of the tree, similar to the animal taxonomy example described above.

It is very difficult for people to interact at an abstract level, although it is much more efficient; we feel more comfortable dealing with more tangible and concrete objects that are represented by the lower levels of the knowledge tree. For example, it is easier for a child to understand what a whale is because it is an animate object which he/she can visually identify with as opposed to understanding what a mammal represents because it can mean so many different things in the animal kingdom. A mammal is just a word to encompass all animals that are warm-blooded; it has no physical representation.

Design team are now commonly composed of people from different disciplines and expertise. Each person is an expert in his/her own area, corresponding to a knowledge hierarchy tree. In order to succeed, they must operate at a higher level because it is inefficient and will be difficult to communicate with each other on a very low and detailed

level, one that is specific to their trade. Therefore, it is common for design to start on a high level of the tree and proceed downward; the objective of any collaborative design is to eventually produce a “concrete object”. Obviously, to talk and operate at a higher knowledge level would be an ideal situation, negating having to talk on a very basic and low level. In order to communicate together effectively at a high level, there must exist some sort of overlap between the knowledge tree of the two people which in turn shows the commonality of knowledge between them as illustrated in Figure 2:



**Figure 2: Knowledge Overlap**

The higher the overlap exists, the more of an opportunity exists that two people can communicate at a high level. The next chapter will describe how groupware can “bridge” this knowledge between individuals.

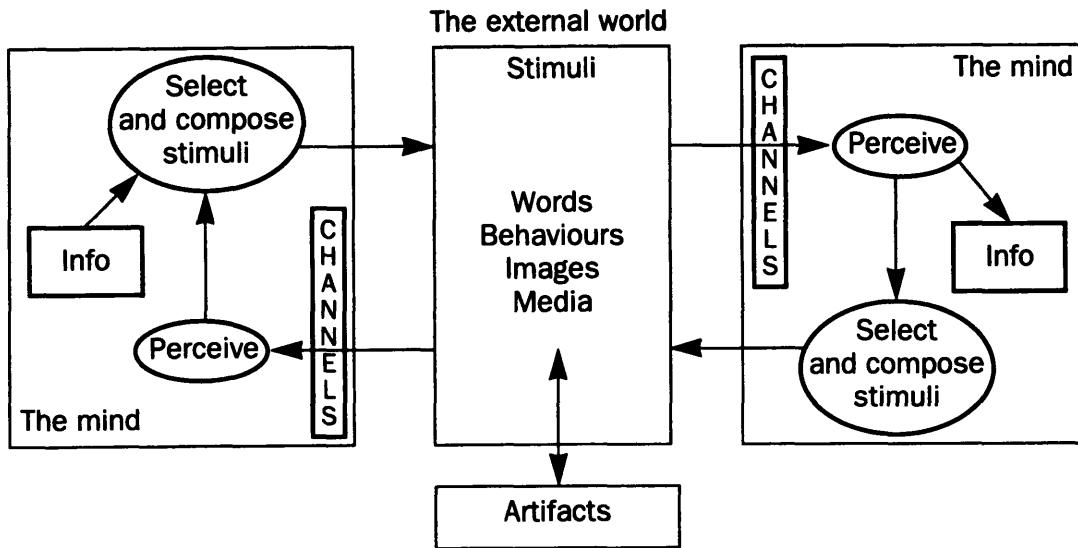
## **Chapter 3 - Groupware**

### **3.1 Definition**

Recently, there has been a lot of discussion about groupware and its implications in the business place. But what is groupware? Groupware is defined as computer based systems that support groups of people engaged in a common task (or goal) and provides an interface to a shared environment [Ref. 2]. Note that the definition above does not specify that the users be active simultaneously. There exists groupware that support either synchronous(i.e. video conferencing) or asynchronous(i.e. email) modes of information exchange. Three concepts are critically important in trying to understand the issues of groupware: communication, thinking, information availability.

### **3.2 Communication**

Obviously, a group must be able to communicate in order to achieve its goals. Understanding the communication model is important in evaluating the importance of dynamics of groupware. Communication is usually a multiparticipant process. For example, a person will come up with an idea from some stimuli and wishes to communicate this to his peers. He first has to choose some combination of elements (i.e. words, images) and then to frame them in some medium like paper, voice, or video. The coordination of these stimuli will impact the way the communication is received by the listener. However, the mental model developed by the listener from the stimuli may or may not match the original idea of the initiator.



**Figure 3: Communication Model [Ref. 3]**

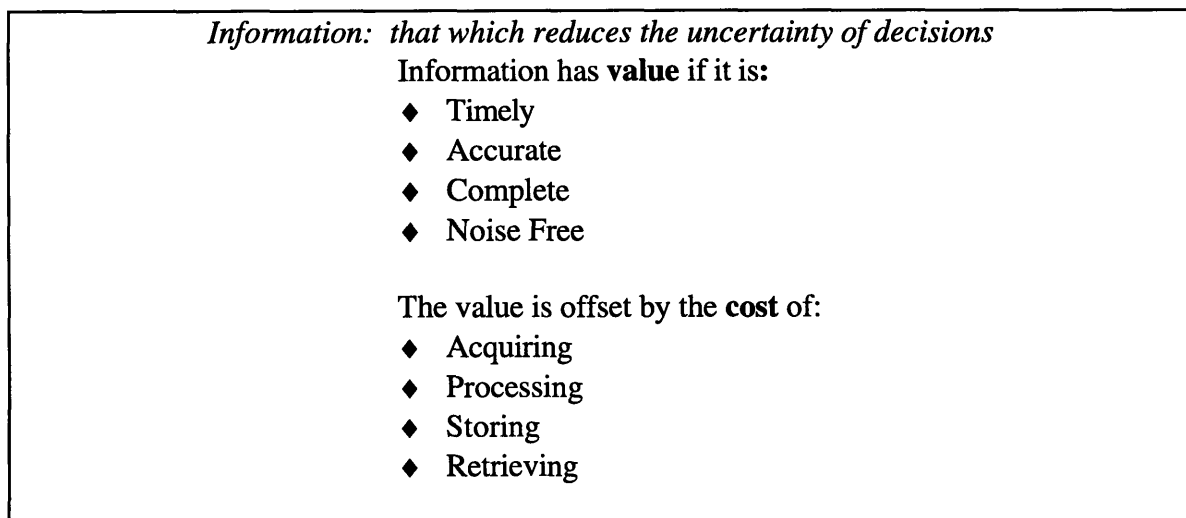
By operating at a very low and concrete (detailed) level, one can more or less communicate on a 1:1 basis, meaning that the mental model of the listener will match that of the initiator. But of course, this is a very inefficient way of communicating.

### 3.3 Thinking

When coordinating and exchanging information in a group, distractions become commonplace. It is important for groupware tools to help alleviate these distractions and to focus group activity on the relevant issues at that moment, and hence improve productivity. Electronic meeting tools can help focus and support group thinking. An evaluation tool might help a group's thinking on the possible outcomes and cause certain assumptions and values to be made more explicit. Groupware is not merely a tool to help facilitate information flow; it must function as a organizational information tool.

### 3.4 Information Availability

Information can be defined as anything that reduces the uncertainty of the consequences of choosing one thing over another. Groups have to be able to exchange information in order to achieve their goals, and it therefore follows that one needs the appropriate and correct information to achieve these means. Figure 4 depicts the model of information:



**Figure 4: Information Model**

Information has value only to the extent that is timely, accurate, complete and noise free:

- Timeliness means that the information is available when the choices are made.
- Accuracy means that the information reflects true state of the world.
- Completeness means that all elements that would affect the outcome are present.
- Noise free means that the information is relevant.

The cost of retrieving, acquiring, storing and processing the information may outweigh the value of actually having it. Consider the following albeit extreme example. A company could store important information by writing it on a piece of paper and then depositing or dropping it in a container. When someone wants to retrieve some information, the cost of trying to find the information would render it so high as to be useless, even though

everyone knew where to look for it. With the increasing flow and storage of information, it becomes increasingly important to retrieve it in a cost efficient manner.

### 3.5 Time-Space Taxonomy

Groupware can be thought of as a tool to help a group that is congregated in the same room or distributed in many geographic locations. It should also be used in either an asynchronous (non real-time) or synchronous (real time) interaction. This relationship between time and place is an inherent quality of groupware that can be summarized in the 2x2 matrix shown below in Figure 5:

	<b>Same Time</b>	<b>Different Times</b>
<b>Same Place</b>	Face to face interaction	Asynchronous interaction
<b>Different Places</b>	Synchronous distributed interaction	Asynchronous distributed interaction

**Figure 5: 2x2 Time-Space Matrix**

Some have argued that the shaded area (Different Times/Same Place and Different Times/Different Places) of this matrix and that the distinction between these two cells is blurred; they should instead just represent one box (eliminate the dotted line) and merely state the general asynchronous tools. By default, different times implies an asynchronous interaction and it doesn't really matter where the other user is located.

Currently there are many tools to support these various time-space interactions. A Same Time/Same Place interaction would mean all parties are located in let us say a common room and thus computer meeting tools would be used to help facilitate the meeting like keeping track of agenda and meeting minutes. For the Same Time/Different Places interaction there exists video/audio conferencing where each location is hooked up

with a video-phone device where by all parties can view and see each other despite being in multiple geographic locations. Since this is a synchronous tool, it does require everyone to be present at the same time; this makes it difficult to conduct international video conferences where one side will be at the other end of the time zone. The Different Times/Same Place and Different Times/Different Place interaction is probably the most popular aspect of groupware because of the attention it has garnered through the press. Asynchronous tools that exemplify this interaction include e-mail, Lotus Notes, the Internet and the World Wide Web (WWW). The great advantage of these tools is that anyone anywhere in the world with the proper connection can view and use the information at any time. With this ability, people are no longer constrained to working physically side by side with another team member. It has essentially redefined the work model whereby everyone has to go to the office to work. These tools force the power back to the workers whereby they can now get their needed information at any time of the day and at anywhere in the world. This fundamental shift will hopefully allow workers to be more productive and efficient in their workplace.

### **3.6 Software system**

But what is it that differentiates a groupware versus other software systems? A groupware system helps support *groups* of people toward a collective goal. Contrast that with most software systems today which only support the interaction between the user and the system. Within a company, these systems can be characterized as either personal productivity applications or operational systems. Personal productivity applications focuses primarily on individual tasks such as writing a letter or preparing a spreadsheet

analysis. Operational systems manifest themselves as a centralized system conducting such back office functions as payroll or ledgers. Groupware systems break this sort of mode and incorporates a decentralized model of computing; empowering distributed teams to work collectively as a group. Lotus Notes, a groupware software system developed by Lotus Development Corporation., utilizes the features mentioned in this chapter to make it stand out as one of the premier collaboration tools on the market. Chapter 4 will take a more in depth look at Lotus Notes and its key components in incorporating groupware features and functions and how it attempts to solve some of the problems that face collaborative design.

## **Chapter 4 - Lotus Notes**

### **4.1 Overview (Notes as a Strategic System)**

Lotus Notes, a strategic system application which helps people work together to accomplish certain goals, is a communication-centric system which uses documents as its currency and helps automate some of the business processes. A strategic system is designed to support the “extended workgroup” because workers now are no longer static and working within the confines of a particular company. Not only are workers more mobile, but teams also incorporate a broader collection of people from different departments and therefore requiring a communications platform able to meet the needs of all team members.

Communication thus becomes a critical part of groupware. Sending and sharing of data and among workgroup members is very limited in operational and personal applications. Success of groupware systems is dependent on users sending and sharing information in a document repository. It allows workers to address a business process from start to finish, monitoring its status as well as serving as a “corporate memory” repository where experience, expertise and unique business knowledge can be captured and recorded for an organization. However, one of the most important features in Lotus is not its document management features, but the security and replication characteristics that makes it a more robust and efficient groupware tool.

### **4.2 Document Database**

With a multidisciplinary team working on a collaborative design project, it would be safe to assume that each discipline works with their own type of software application

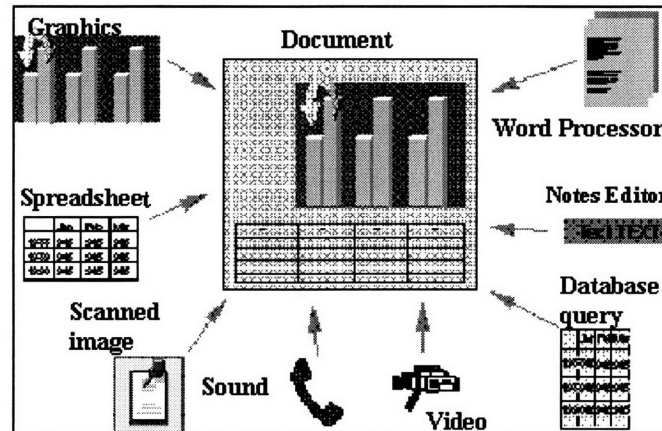
package. For example, an architect might use a Computer Aided Design(CAD) tool while an engineer uses a finite element analysis package. There needs to be some sort of medium from which output from these different packages can be gathered and sent electronically so anyone can view and use it at any time. Lotus Notes is a tool that combines all these elements into one document capable of combining these different types of data.

Lotus Notes is a document database and the basic element in a Notes database is a document. The structure of a Notes document is defined by a form which contains a number of fields. For example, an insurance document may contain fields such as date, policy number, name, address, text of policy and so on. Notes allows users with special access to be able to change and tailor these fields in order to suit the specific tasks of the workers.

Because a document may contain many fields, Notes allows users see the document through “views”. When Notes presents a view to the user, it will list the field names in columns. For example, if a user wanted to look through the documents by dates, then Notes presents a view of the documents sorted by values entered in the date field. Other field information could be listed in columns to the right of the first column. This feature allows users to specify their own “views” on how they want to sort or organize their documents.

Although the basic element of a Notes database is a document itself, a Notes document can have an arbitrarily unstructured content. Because it is able to store and manage collections of data types such as tables, formatted texts, graphics, audio, video, scanned images and linked/embedded objects, it does not readily lend itself to customary

relational or other database systems. Fig. 6 below depicts the various types of multimedia objects and data types that a Notes document can incorporate:



**Figure 6: Notes Document**

With such a multitude of documents being shared and edited by many different people, Notes utilizes a feature called version control which provides document versioning capabilities in order to track these changes. Automatic versioning is enforced in forms where each edit is either posted as a main document or as a response to an original. An important use of versioning control is to document and keep track of the primary contributors to each project and to encourage everyone to participate during the life cycle of a project. To truly utilize Notes to its full potential is for everyone to use it extensively to exchange ideas, thoughts and their work.

### **4.3 Replication**

One of the downfalls of working with databases is that one has to be constantly connected to it in order to gain access to the information. This is the only way to enter data and have it be updated immediately. As is the case, many people will never use an entire database. They may only use a very small percentage of the data in it, yet logging

into the entire database may waste precious resource for someone who may need to access more information in it.

The essential characteristics of groupware platform is its ability to share information with all members of a group, regardless of time and location. One of the strongest features of Lotus Notes in particular allows people to replicate or “copy” a database and store it on a local server while still maintaining information synchronization with the original database server.

For example, replication technology will allow users at a remote site to make a “copy” of a database in let us say New York and store it on their local server. Now, rather than requiring that users in the remote site connect to New York to access the database, replication will allow the workgroups to keep information synchronized across geographically dispersed sites at their own local server. It allows workgroups to specify which databases they want to replicate locally, since they may only need a portion of the entire database. This technology makes it a very attractive solution because now everyone does not have to dial in and access the same server, thereby reducing the communications cost and network traffic. Some of the key features that characterizes Notes replication technology are listed below:

**Bi-directional** - Any changes to documents in a database occur at any or all sites that use the database. Notes has the ability to accommodate all changes, not just changes that are made at a central site to be propagated to the remote servers.

**Efficient** - When synchronizing geographically dispersed databases, only those documents that are new or have been changed on either side of the replication need to be replicated. Notes recognizes the appropriate documents and uses the minimum resources to conduct the task.

**Client Replication** - Occasionally connected workgroup members such as mobile users working at a remote site need the same level of access to server databases as connected

users. Notes is not limited to server-to-server connectivity, but also includes client-to-client connectivity.

**Selective Replication** - Notes allows users to define the profile of documents that they need to replicate to their client workstations. Using selective replication, users can replicate only those documents that have changed within a predefined number of days or those authored by a particular group member.

**Background Replication** - Notes replication can run in background mode, allowing the user to continue to work on other tasks.

With the ability to selectively replicate databases to a local server, security becomes a major concern for a company planning to implement Notes. In the next section, security issues will be discussed and how Notes deals with this highly important topic.

#### **4.4 Security**

Today's competitive business environment requires organizations to break down traditional barriers within their enterprises in order to foster innovation, creativity and increase productivity. With the multitude of documents generated by a workgroup or a company, it is critical for groupware systems to incorporate security features not only to prevent outsiders but also to prevent people within the company to view sensitive documents that they do not have clearance to look at. Lotus Notes presents a very advanced security system based on security access levels and encryption to protect documents from unauthorized visitors.

The key to security in a distributed system like Notes is by using encryption. The *de facto* standard for access to X.500 directories is the X.509 certificate, which is based on the RSA public key encryption technology. Traditional cryptography is based on the sender and receiver of a message knowing and using the same secret key: the sender uses

the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as secret-key cryptography. The main problem is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, or a phone system, or some other transmission system to not disclose the secret key being communicated.

Public-key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman in order to solve the key management problem. In the new system, each person gets a pair of keys, called the public key and the private key. Each person's public key is published while the private key is kept secret. The need for sender and receiver to share secret information is eliminated: all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal. Anyone can send a confidential message just using public information, but it can only be decrypted with a private key that is in the sole possession of the intended recipient.

Using public key encryption and other security features, Lotus Notes provides four levels of security: authentication, access control, field-level privacy and digital signatures.

- \* **Authentication** - Authentication is the means by which a user is reliably identified; that the user is who he/she claims to be, when seeking access to a Notes server.
- \* **Access Control** - An access control list (ACL) regulates who has access to which resources, and what sort of access is allowed (i.e. compose, read, write). Resources under ACL control include servers, individual databases, documents, and fields. There is in turn various ACL levels:

**Manager:** This person owns the database; can add and delete users from ACL.

**Designer:** This person can update or modify the design or layout of database.

**Editor:** Editors can read, write, and update documents in database.

**Author:** Authors can create new documents and can read others but cannot modify documents already in the database.

**Reader:** A reader can only read documents in the database.

**Depositor:** A depositor can only write to a database. This is useful for mail-in databases, like ballot boxes.

- \* **Field Level Encryption** - A user may need to share field-level information in a document with another user while ensuring that no other users can view it. Access control can restrict field-level access only to categories of users (managers, readers), but not to individuals. However, if one should want to only share the information with a specific individual, then a Notes database designer could encrypt the sensitive information in that field so that only a certain person can read it.
- \* **Digital Signatures** - A user may have to verify information that they receive actually was sent to them by the sender listed in the document and that none of that information has been tampered with. Verification is managed by digital signatures created by the public key encryption.

Notes is not the only or should be thought of as the only groupware tool in the market.

One of the advantages of using Notes is that it already has a very large installed user base, for example like Andersen Consulting, so support and documentation will always be there.

However, Notes is a proprietary product whose limits are defined by companies who can afford to install it and propagate it to its employees. New technologies like the Internet are threatening to erode the market share of Notes.

## **4.5 Future**

The proliferation of the **World Wide Web (WWW)** has definitely taken a piece of the groupware market that Lotus had held so firmly for the past couple of years. As with Notes, the WWW allows users to share information with other people but lacks the powerful replication engines to coordinate simultaneous servers. Although the WWW is excellent in terms of publishing information (i.e. price catalogs, quotes), it lacks the feature to look at a document and then edit or respond to it other than by email. Notes allows one to reply to documents and to keep a log of changes made and by whom.

Security and levels of access are critical components which the WWW still needs to address. However, with the low cost of setup and an open architecture as compared with the proprietary nature of Notes, many companies are either switching or experimenting with the WWW as a possible tool for coordination and information dissemination.

Lotus, sensing the impending nature of the WWW, has incorporated the WWW into its latest release of Notes which allows the user the ability to publish Notes data and documents onto the web. Ray Ozzie, CEO of Lotus Development Corporation, has said that in the future Notes will be able to replicate directly to external replication engines of SQL databases and the WEB.

There will also be a fundamental shift in the Notes strategy. Notes is a “document and communication centric” application where many types of media and text can be incorporated into a single document and sent to other Notes client. However, Ozzie has said that the next upgrade will be “occupation centric”[Ref. 4]. Notes will provide an alternative user interface for different occupations, such as a personal digital assistant-specific interface for a mobile user and a different type of interface for a salesperson. One is seeing a concerted attempt to decentralize the process and to empower the user with the tools he/she needs to do a job a more effectively. Instead of having a set standard interface which everyone is forced to use, Lotus has realized that in order to gain or even maintain market share it must target and recruit new user groups who may have before been hesitant to use Notes. By tailoring interfaces for specific groups, they are hoping to appeal to users who may now see what Notes has to offer specifically to them.

## **Chapter 5 - Autocad**

### **5.1 Introduction**

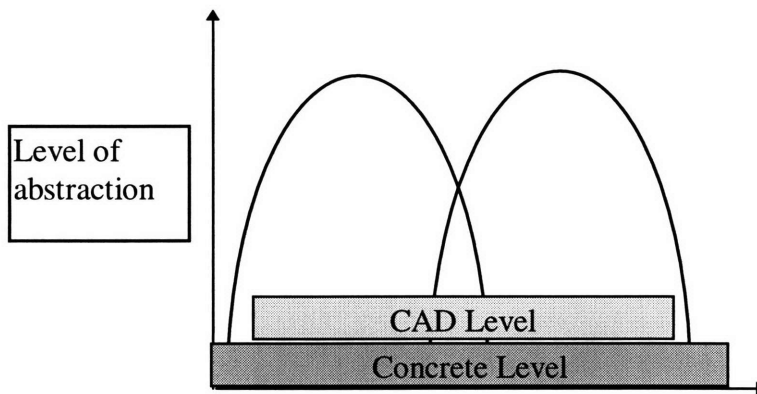
Autocad is an interactive drawing system developed by Autodesk Inc. that is designed to permit a user to construct or edit a drawing on a desktop computer or graphics workstation. To this extent, it is analogous to a word processing program except now the object being produced is a drawing. It serves as an engineering tool designed to capture a representation of reality. Autocad R13, currently the latest release, will be the version that will be discussed in this chapter. Autocad has evolved from just a basic 2 dimensional drawing tool to one that now incorporates 3 dimensional rendering, layering and database technology to facilitate the modern design processes.

### **5.2 Abstraction**

People have a tendency to more easily identify with and understand physical objects as opposed to some abstract idea or concept. A physical object is something that one can touch or feel and be able to associate some tangible meaning to. This is the reason why CAD tools such as Autocad are so powerful; it provides a graphical representation, be it 2 dimensional or 3 dimensional, of reality. Although the object is represented in a virtual environment (i.e. the computer), nevertheless it depicts them as close to reality as possible, without it being there physically.

In essence, as described in Chapter 2, the lower level of the knowledge tree is where the “concrete” level resides (the physical and tangible objects) as illustrated in

Figure 7:



**Figure 7: CAD Level Abstraction**

Figure ABC illustrates how a CAD system fits into the overall scheme of a knowledge tree and its placement in relation to the levels of abstraction. Unlike the concrete level, the CAD level represents a higher level of abstraction, that of representing reality on a computer system but still something which is discernible as a low level object. That is why CAD systems are so powerful because it has the ability of conveying a picture to someone.

For example, if a client wishes to see what his/her house will look like after it is built, an architect has several options to pursue. One way is to verbally describe this house; this is not only very inefficient because it may lead to many misinterpretations of certain words and ideas used by the architect to mean one thing but understood to mean something else by the owner. The best way of course is to provide some kind of visual representation of this house and therefore eliminate much of the confusion or misrepresentation that may occur through just verbal communications. Drawings or physical representations can give more meaning to an object than just words alone. However, the most visually effective way would be to physically build a scaled down model of the house. This is obviously the most ideal situation but it will not only take up precious time in building it but also makes it difficult to modify if there is something that

the client wishes to change. Another way of representing it is to use a CAD model. Although not a physical model as described previously, it does provide 3 dimensional viewing and becomes a more effective tool in aiding the client see visually what is to be built. Plus, it allows changes in design to be quickly made and then rendered immediately to show the effects of these new alterations.

### **5.3 Graphical Representation**

One of the main aims of any design software is to graphically represent an object to a high degree of specificity and all the while developing ways to let the user “navigate” through these representations. People have always wanted more from CAD systems than just being able to just “draw” something. CAD systems used to be only able to represent a 2 dimensional plan view of objects and that was all. CAD systems had to develop more meaning and structure and evolve itself into much more of a design tool, rather than some application which has such a limited scope. Current CAD systems represent more than just being able to visually replicate some object on a computer. It now allows designers to not only represent objects in a 3 dimensional environment but also to graphically check connections and fittings (i.e. pipes, walls).

One of the most celebrated cases on how a CAD system was used in this capacity is that of the design of the Boeing 777 aircraft [Appendix 2]. The design of this plane was done entirely with 3 dimensional CAD technology. This leap to 100% digital design allowed Boeing engineers to view, design, and pre-assemble - at any angle - the entire plane and its more than 3 million (132,500 unique engineered parts) on screen. [Ref. 5].

In order to manage such an enormous set of drawings, one important feature of an Autocad system is that of layering. This means that any drawing entity can be assigned to any layer of the drawing. This allows separation of portions of the drawing. For example, all brickwork could be assigned to one layer, while all electrical work can be assigned to another. Layering can also be used to an advantage while editing a drawing. For example, suppose a plan drawing is almost complete with only the brickwork cross-hatching to go. If all the brickwork is in one layer, then each other layer can be *turned off* so that only the brickwork is visible and selectable. That makes it much easier and possible to cross-hatch the brickwork without inadvertently affecting something else. By integrating a layering system, it allows people to work on different parts or layers of the design drawing and then collate them together.

What was once a drafting tool to draw planar or 2 dimensional views now supports 3 dimensional modeling and rendering. Some may wonder why even bother with 3-D when all the contractor needs is a 2 dimensional drawing to construct the building. One of the most important concepts of business is to continually add value to your products. Autocad has therefore modulated or evolved itself by implementing 3-D not only as a marketing tool but also as a means to more accurately spot problems. A 3 dimensional view presents a more realistic picture of an object or building, especially to a client who is not familiar reading the (2 dimensional) plans of the building. This view also allows an engineer or architect to more accurately depict and spot problems concerning how each component interacts with each other in a 3-D space.

## 5.4 Object Database

In a collaborative design process consisting of multidisciplinary team members, not everyone has a specific interest in the entire CAD drawing. Some may only be interested in the brickwork, while others are interested only in the design of the windows. Given a complex building or project with millions of parts, there needs a way to be able to track not the individual lines or circles but the “objects” that these elementary shapes create. For example, one might be interested in keeping track of a window object which was constructed from multiple straight lines and shapes in CAD. Because it is now an object, it must also contain some inherent properties like color, opacity, etc. When an architect wants to create another window somewhere else in the building, all he has to do is call this window object and it will contain the same properties as the one before. By creating a large database of objects, one has the ability to more easily track and manipulate them as well as enabling other applications or contractors to work on only those objects they are familiar with.

One can now appreciate how teamwork and collaboration must play in this aspect of the design process. The architect does not have an in-depth knowledge of the properties of every type of object in the house and an engineer or contractor may only be specially trained in a few of these aspects. The architect or engineer will have to work concurrently with multiple contractors who may help them define the properties of a window object, door object, etc. By engaging themselves early on in the design process, they eliminate any surprises or misunderstandings that may occur later.

Autodesk has reengineered the Autocad DWG format into an object-oriented database (AcDb). This architecture provides a powerful tool for third party developers to

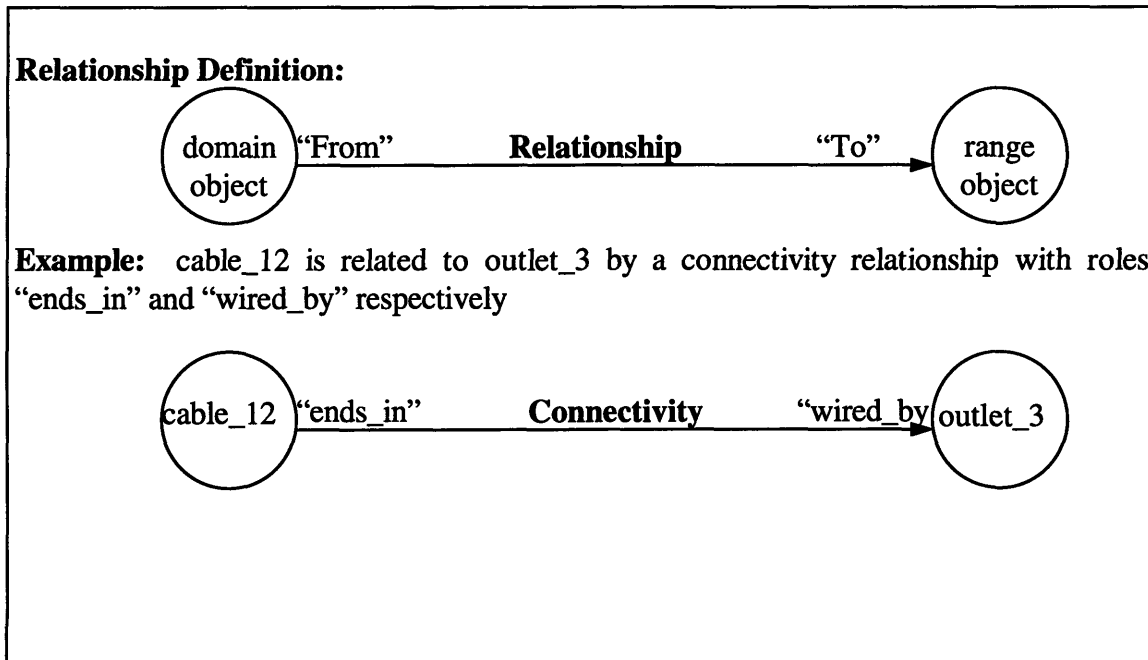
build integrated applications more quickly than ever before. It provides the foundation for development of seamlessly integrated and sophisticated extensions to Autocad. Now, other applications can have access to these objects stored in the Autocad database. It also allows objects created from independently developed applications to appear as though they are native to Autocad and can be incorporated into the DWG file [Appendix 3].

In addition to the database described above, Autodesk has developed a prototype software library (IFCProto2 also known as Babylon) that enables developers to create AEC (Architecture Engineering and Construction) object databases that can be shared with other applications in the industry. It provides a more powerful mechanism for creating, sorting, retrieving, navigating and browsing AEC information represented as objects, relationships, and other object attributes.

Any applications that will use this new library class will be able to do any of the following 3 functions:

1. Create an object (associating attributes and relationships) and store it in a database.
2. Retrieve an object, access its attributes and relationships and modify them if necessary.
3. Remove an object from the database.

One of the interesting aspects of Babylon is that it allows developers to create “general” relationships between objects:



**Figure 8: Example of IFC Relationship**

A relationship is identified by its type, the type of objects that it relates and the roles in which the connectivity plays in the relationship between the objects. Now, one sees the evolution of how objects have become more than just a single unconnected entity. By imbuing this type of function within an object, it makes it more intuitive and meaningful in its overall context of the entire project.

As in all databases, this Babylon library provides a query mechanism which allows developers to search through the database to retrieve certain objects. The current query mechanisms include: 1) Extent query: returns all objects that are instances of the query input 2) Relationship query: Given an object Z, returns all the objects of a specific type that are related to Z via a certain relationship. For example, one could use this query mechanism to calculate the total length of all the walls made out of brick or view all objects that have a relationship with a shear wall.

## **Chapter 6 - Case Study**

### **6.1 Introduction**

For the past couple of years, several departments at the Massachusetts Institute of Technology have embarked on the Design Studio of the Future project, an interdisciplinary effort to focus on geographically distributed electronic design and work group collaboration [Appendix 4]. The project integrates advanced computer and telecommunications support to allow new ways of providing access to knowledge and structuring collaborative processes. The design of a building or any structure basically simulates that of a virtual organization; collaborators such as architects and engineers will come and work together for the life of the project and then disband so that the components can be recombined for another new project. With the growth of global networks and telecommunications, it is now possible to incorporate and integrate geographically distributed multidisciplinary teams to form this virtual organization. No longer must everyone be at the same place to work. One of the goals of the Design Studio is to explore these collaborative issues and how it relates to the design process.

At the Massachusetts Institute of Technology during the Spring 1996 term, a collaborative design studio course was offered jointly by the Department of Architecture, Civil and Mechanical Engineering (1.964 - Special Studies in Civil and Environmental Engineering, 2.998 - Advanced Topics in Mechanical Engineering, 4.132 - Level II Architectural Design). Multidisciplinary teams (usually consisting of two architects, one civil and one mechanical engineer) were formed to develop design proposals and mockups for projects which required the team to incorporate the user's needs, aesthetics and to

meet performance criteria (i.e. structural stability). The following sections will focus on the use of technology between the group members and how the design process affected the overall team dynamics.

## **6.2 Technology Use**

Each team was provided with the necessary software applications such as Lotus Notes, Autocad, SAP 90; there was no explicit instruction that one had to use a specific application. It was up to the teams to decide what they wanted to use, which may have been one of the reasons why some groups used Notes extensively and others didn't.

One of the reasons why there was such a disparate usage of Notes by different teams was that Notes was not installed on the UNIX platform, the platform which Athena, the central computing environment of the Massachusetts Institute of Technology, runs on. Students use Athena extensively for everything from email, World Wide Web browsing, writing papers and computer programs. For some teams, it was just inconvenient to "email" another team member in another computing platform like Notes because one would have to assume that the other person would log into Notes each day to read his mail. However, it is safe to say that students log into Athena at least once a day, if not more, and that provided a more convenient and efficient way of communicating with each other for some teams. Although Notes does allow one to send email to other Internet addresses outside of Notes, at the time this was written, the capability to do this was not yet implemented.

Another reason does not concern the technical side but more the structure of the course. Since the professors allowed a lot of freedom in allowing each group to decide

whether to use Notes, this might have lead some teams or group members to be hesitant in trying to learn or understand a new database/communication groupware application if they were not forced to learn it. Compare this to a corporate environment where one does not usually have a choice in deciding which communication/database management tool one wants to use. In order for collaborative design to work effectively in a geographically dispersed or multidisciplinary environment, there must first be some kind of communication standard set to send information back and forth, be it a groupware application or even use of the Internet.

However, for a team that did use Notes extensively, a student indicated that it was a very convenient and useful tool in collecting and distributing information. By using OLE, he was able to easily embed Autocad drawings within the Notes form and then send it off to another team member who was then able to either to view or edit it. Another useful feature was version control, whereby one could keep track of who last edited the drawing and when it was altered.

During a design review session, conducted via video conferencing, with design professionals from Sverdrup in New York who also had Notes, the team was able to communicate their drawings through Notes to these professionals and this added a level of convenience that was never possible. This enabling of correspondence without having to fax or ship a document overnight to a remote site saves everyone time and money. People were able to edit the drawings and see the results on the other end in real time, enabling far more of a “physical” interaction than ever before.

### 6.3 Interaction

Technology only represents one part of the design process. Perhaps a more important aspect is that of the interaction between the multidisciplinary team members consisting of architects and engineers. One of the interesting aspects that came up is how different objectives emerged in one team.

In this team, one person was more concerned in just finishing up this course and graduating and not really devoting a full time effort to the project. Another team member was more intrigued with the computers and technology that was present and wanted to just make something interesting with all computer equipment available. Although there was a main goal of eventually producing a workable design, each person's individual agenda may have inadvertently affected how successful the team was going to operate and interact.

During the first project of the semester, each team member was more likely to contribute to the design of the product, in this case a chair. Yet this "design by democracy" was successful only in allowing different disciplines to participate initially; in the end it just created more designs to go over and critique and eventually proved too time consuming. Unfortunately, this process then reciprocated back to where the architects would design something, with or without some input from the engineer, and then give it to an engineer who would then evaluate it in terms of feasibility and stability after which it would then be returned for any modifications and so on.

The collaboration between the architects and engineers has always been one where one tries to coerce the other to accept the other's viewpoint. This was the exact same case here where although the end result was a chair, the means of getting there took

different paths for the engineers and architects. The architects wanted to create not just another ordinary chair but something unique yet conform to some precedence from chairs in the past. It was important to them that this chair correspond to some sort of metaphor (like movement, space, etc.). For some of the engineers, it came down primarily to feasibility (i.e. whether the user would feel comfortable sitting in the chair) and structural stability. They themselves were more interested in creating something that would work and be functional to the user. These attitudes, which have been prevalent for a long time, perhaps still roots itself in the educational process. Engineers have always dealt more with numbers and equations and architects more with history and art. I believe that the only way for any of these multidisciplinary team to work cohesively and effectively is for everyone to be open minded for suggestion and embrace each others' strengths. Without this openness, one will come upon this problem forever, and hopefully this Design Studio class will somehow provide the environment and tools to foster this kind of collaboration which will be necessary in a more competitive market. In the future, each discipline will need to harness the skills of the other for the success of the project.

## **Conclusion**

With the advancement of tools such as Lotus Notes and Autocad, engineers have been given the opportunity to collaborate on design projects anywhere in the world and at anytime. The expanding global economy and competitiveness has made it even more important that the best people are sought for in each project; this makes it critical for groupware applications to help bridge the communications gap for team members in geographically dispersed sites. Although Lotus Notes is by no means the only groupware application out in the market, it does provide an integrated and secure environment for exchanging documents. The rapid growth of the Internet will provide a real test of how Lotus will try to keep its competitive advantage in the workplace when companies are now seriously looking at deploying intranets for coordination and dispersal of information.

The use of Autocad and other CAD systems have enabled engineers to visualize their work and provide a more deeper and better understanding for clients looking at the drawings. Drawings themselves provide a powerful visualization tool at a low enough abstraction level that anyone can understand. With the incorporation of 3-D modeling and an object database, Autocad has positioned themselves not as just a drawing tool but as an information tool where now lines and shapes contain a new dimension, that of incorporating information into objects. This helps in not only navigating and querying certain types of objects but also allows other AEC developers to export/import these objects into their own applications.

As discussed in the last chapter, technology represents only one component of the collaborative design process. Working in a multidisciplinary team requires a certain

amount of structure and for each team member to put the goals of the team ahead of those of the individual. The Design Studio class has provided a rare opportunity and insight for students from different academic backgrounds to work together in a project. It will be inevitable that this kind of organization will become more common as communication costs go down and the need for a broader breath of understanding for engineering projects increases.

## Appendix

[1] OLE stands for **Object Linking and Embedding**. It is a Microsoft Windows technology that establishes a standard way for applications to communicate with one another. The advantage is that one can combine, process, and share information from different applications (represented by objects) in a single source. The source that contains the object is called a container.

An OLE object is a software representation of a physical or logical entity, for instance, a calculator or chart. It is essentially an application that can only run in the context of a container. It carries all the information it needs to function but needs a container to plug into.

Objects have public and private elements. The private side involves the details of how the object is implemented. The public side consists of a set of programming functions that other applications can call in order to set and retrieve object attributes, and to manipulate the object. With OLE, you can either link or embed an object in a container. If an object is linked, it is visually represented in the container, but the software that allows an application to edit the object resides elsewhere. This means that, although updates to the object appear in the container, the updates must be made outside of the container, in another application window. If an object is embedded, the entire object—its physical representation and editing capability—resides in the container. In this case, updates to the object take place within the container. The object does not save its data directly to disk. OLE controls are always embedded.

One of the OLE features is in-place activation. With in-place activation, you can double-click an object in a container and work with data in that object. You do not need to switch to another window or application. When the object has control, its menus, palettes, and controls replace those of the container application. When the container application regains control, its menus, palettes, and controls are restored. In-place activation offers you the ability to perform operations for multiple applications from a single location. In so doing, one can focus more on the tasks you need to accomplish, rather than the applications required to accomplish them.

[2] The new Boeing 777 measures 209 feet long, can carry 440 passengers, and carries a price tag of \$120 millions to \$170 millions. The commitment to a total 3-D digital design meant that not only new technology must be incorporated, but also a fundamental change in the way Boeing did business internally, with its supplies, and its customers.

“We came up with this concept of a design/build team where all the planners, toolers, and engineers work as a team, sit together, and design the airplane so that you can cut down change, error, and rework out of the factory.”

-L.M. Olson, Director of Information Systems for the 777

A total of 2,800 engineers accessed 1,700 workstations worldwide for this project.

By using the CAD systems and 3-D solid modeling, Boeing was actually able to attain a 60% - 90% error reduction in its design. It not only reduced the development and manufacturing life cycle, but also improved the quality of the designs themselves (the first 777 was just 0.023 inches away from perfect alignment). Digital design has eliminated the use of mock-ups during the 777 design process and has greatly enhanced the interface identification. In fact, the system found more than 10,000 interfaces that would not have been caught until after assembly.

[3]

**The following comes from the homepage of:**

**<http://wwwwis.cs.utwente.nl:8080/~faase/dwg.html>**

## **Autocad DWG format**

AutoCAD uses the, so-called, DWG format to store its drawings. The specifications of the DWG format are not disclosed by AutoDESK, and it is known that it changes for each new release of AutoCAD. Up to release 11 the format has undergone many small changes, usual extensions. With release 12 the format did not change, but this was only the calmness before the storm, because the format of release 13 is a complete new format.

To allow the exchange of drawings (or parts of them) AutoCAD has introduced the DXF format. DXF stands for Drawing eXchange Format. The specification of the DXF format can be found in each set of documentation with the AutoCAD program. Although the DXF format is intended to be completely public, this is not the case. With release R13, for example, this includes all the ACIS information related to solid modeling. The reason is obvious; ACIS was bought by AutoDESK, under a non-disclosure agreement.

The DXF and the DWG files essentially contain the same information. The DXF format contains all the essential information found in the DWG format to represent a drawing. The DWG format does contain some extra information related to internal book-keeping which is needed, for example, for the PURGE command.

Although the DWG format is not public, many people have tried to decode it, and use the results in commercial software, either in converters between the DWG format and other formats, or in programs that allow a DWG file to be viewed without AutoCAD (even as Netscape plug-ins). There are also companies that have made libraries available to read and write the DWG format. (Reading is easier than writing due to the many check-sums that are stored in the DWG format, which are hard to crack.)

## **Information about the DWG format**

Ilija Kovacevic has released a (once commercial) source library in C for reading and writing DWG files for non-commercial use. It supports Versions 2.6, 2.6, and Release 9, 10, 11, and 12. For more details see his software page.

I too have worked on decoding the R12 DWG format. Some others have contributed as well. Our results are available by means of a C program, `scan_dwg.c` (Reini Urban has a more extended, but less bug-free version), that is available under the GNU license. I have started with a description of the format using a Binary File Format specification language, which is rather incomplete at the moment.

## Conversion between different releases

As a rule each new AutoCAD release can read DWG files of earlier releases. AutoCAD R13 can write DWG files of R12. If you need a stand-alone utility to convert DWG and DXF files to earlier releases, try **BackDraft** from MarComp. AutoCAD does provide DXF convertors to earlier releases, as well as to newer releases. *(This is what I can remember. It has not been verified, and might be out-dated.)*

- There is/was shareware called AC11TO10 which would do this directly...despite the name, it works with release 12 also. According to: Wayne Scharf <natron@kaiwan009.kaiwan.com>

There are some problems with converting ACIS solids (in R13) down to R12. They are either converted to a wire frame or to AME solids (using a utility provided with AutoCAD R13). According to: Eric M. Field <emfield@virginia.edu>

## DWG R13

The DWG 13 format is the format used by the latest Release of AutoCAD. It has taken some time before Release 13 has become stable, and it is now being used more widely, and slowly replacing Release 12. The format has changed for each sub-release of R13 (including the beta release). For R13c4 it is going to change again; a minor change this time: ACIS 1.5 will be replaced with ACIS 1.6. (See also: AutoCAD Release 13 Bug List compiled by Steve Johnson.)

The R13 format has a completely different structure than the R12 and preceding formats. Note that the R13 DXF format has also undergone major extensions. Especially, the introduction of the groupcode 100 with "AcDb. ." strings suggests that there are two ways for reading the DXF file. My research on the DWG R13 format, has lead me to conclude that this second way of reading comes closer to the internal organisation of the DWG R13 format, than the traditional one.

I made some attempt on decoding the R13 DWG format (but stopped working on for the moment). I wrote an account of my search. Robert McNeel told me that Cyco International in Holland have reversed engineerd the DWG R13 format. I have noticed that several other companies also have upgraded their products to the R13 format. For this reason, I will start to update the DWG pages to include information about which product support Release 13.

---

## More specific information

- DWG viewing programs.
- DWG conversion and translation programs.
- Applications with an DWG interface.
- DWG libraries.
- About DXF.
- Companies.

**New:** There is Postscript file of the DWG related pages available. (Generated with `html2tex`.)

## Other sources of information

- SoftInfo: try 'DWG'.
  - Registered AutoCAD Developers try: 'Registered Developer, DWG'.
  - CADENDE articles about: DWG and viewers.
  - CAD Translators, Postprocessors and IC Plotting Service.
- 

## Acknowledgements

I would like to thank the following persons for providing information (by sending me mail, email or by posting on the news):

- Dave Barberi <dave.barberi@bentley.com>, Electronic Communications Coordinator from Bentley Systems Inc.
  - Ray Bentley <ray.bentley@bentley.com> from Bentley Systems Inc.
  - Trey Boudreau <trey@gie.com>.
  - Eric M. Field <emfield@Virginia.edu>.
  - Paul Kasanda <spicer@magi.com> the Eastern Canadian Sales Manager for Spicer Corporation.
  - Darrel Knight <darrell@mindspring.com>.
  - K.C. Jones <kc@autodesk.com>.
  - Robert H. Lieberman <rhl@interaccess.com>, President of AutoMate Corporation.
  - Ben H. Loosli <loosli@bbs.ug.eds.com>.
  - Neil Moffat <nmoffat@pcug.org.au>.
  - Todd Moore <tamoore@vixa.voyager.net>.
  - Oettinger <oettinger@aol.com>
  - Jason Osgood <josgood@frame.com>.
  - Joe Ragosta <jragosta@dca.net>.
  - Mohan Reddy <mohanr@infotech.stph.net>, CEO, InfoTech Enterprises, India.
  - Wayne Scharf <natron@kaiwan009.kaiwan.com>
  - Nancy Schneider <jaded1@netcom.com>.
  - Dennis Shinn <dennis.shinn@saug.wa.com>.
  - Akos Szabo <aszabo@globe.graphisoft.hu>.
  - Joseph Thornwall <jthorn@pcug.org>
  - Reini Urban <rurban@sbox.tu-graz.ac.at>.
  - Eric Warshawsky <esw@netcominc.com>.
  - Rod Young <hermit1@ix.netcom.com>.
  - Jan Zmyslowski <ax711@torfree.net>.
- 

*Last update: Wednesday, 03-Apr-96 10:43:43 MET DST*  
*My home page. | Final words . . .*

[4]

The following comes from the homepage:

<http://alberti.mit.edu/dsof/overview.html>

## Design Studio of the Future Overview

- Scenario: New Product Launch
- Scenario: Construction
- Project Description
  - Purpose
  - Support to Date
- DSOF Role
- Program Objectives
- Program Components
- Faculty Leadership
  - William J. Mitchell
  - Woody Flowers
  - John R. Williams

### Scenario: New Product Launch

An international manufacturer of oil and gas drilling equipment is threatened by a new rig control unit just introduced by a competitor in Japan. The manufacturer is faced with the challenge of leveraging the expertise of its core technology development group located in Austin, its product customization and support teams in Tokyo, Frankfurt and New York and its marketing team in London to develop a market response with a competitive product within eight months. Since the local product support and development teams work on multiple products simultaneously, it is very difficult to completely dedicate engineers to specific projects and relocate them on short notice. Since physical colocation of the requisite expertise is out of the question, the company decides to make use of an adhoc collection of communication tools from telephone, fax, electronic mail, and teleconferencing to face-to-face meetings, express mail, and regular mail to help the team achieve the stated objectives. The result is endless phone-tags because of the time difference, misplaced faxes, delayed mail packages, poor communications among team members because of cultural and language barriers and a lack of cohesiveness among the team members. Needless to say, the company missed its product launch target by over 12 months.

### Scenario: Construction

Imagine an architect of the near future. On a construction site for a resort in a remote part of Indonesia, to resolve a problem, she boots up her laptop and opens a videophone window to speak via wireless linkage to her structural consultant in Los Angeles. They share concurrent, real-time, interactive access to the CAD three-dimensional model of the project, and they refer to it as they

speak. (The model resides on a server at the home office in Boston). After some discussion, she decides to bring the client into the conference; he happens to be in a hotel room in Tokyo. The problem that concerns them is one of site access - a landslide has unexpectedly created difficulties in locating a bridge - so she instructs an interface agent to find some relevant satellite images so that the conferees can assess the extent of the damage and consider new locations. As they speak, the agent searches the network, and soon reports back with what is needed. A course of action agreed upon, the architect sketches the solution on the spot and shares it with an assistant in Boston, giving video, text, and voice instructions on working out the details. The assistant modifies the geometric model, and soon electronically transmits change orders to the contractor's head office in Seoul. This work has been conducted in a virtual design studio.

## **DSOF Program Description**

MIT has invested over \$7 million in studio infrastructure, networking, and design and product development research to create a program that can address the most pressing concerns of a global design and engineering corporation of the 21st century.

The DSOF program is now inviting a select group of global engineering and manufacturing companies to join us as program sponsors and partners.

### **Project Purpose**

The MIT Design Studio of the Future is an interdisciplinary effort between the School of Architecture and Planning (including the Departments of Architecture and Urban Studies & Planning and the Media Laboratory), and the School of Engineering (including the Mechanical Engineering Department, Civil and Environmental Engineering Department, the Lab for Computer Science, and the Rapid Prototyping Laboratory) to focus on geographically distributed electronic design and work group collaboration issues.

The visionary leader of the project is Dean William Mitchell of the School of Architecture and Planning, known for creative experimentation with emerging computer technologies in the arena of design and collaborative work, whose concept of the "virtual design studio" lays the groundwork for the effort. The Faculty Chair for the project is Woodie Flowers, Pappalardo Professor of Innovation in the Department of Mechanical Engineering, known inside MIT and well beyond for his dedication to innovative design education and teamwork, and his innovative approaches to product development.

The physical elements of this virtual studio comprise networked computer and videoconferencing connections among electronic design studios at MIT in Civil and Environmental Engineering, Architecture and Planning, Mechanical Engineering, the Lab for Computer Science, and the Rapid Prototyping Lab. This prototype collaborative design network is known as StudioNet (TM). The most recent iteration provides Wide Area Network connections and working relationships with industry partners and sponsors, to introduce real design and construction and manufacturing problems into the equation: corporate partners, faculty, and students can now tackle meaningful design issues in a truly geographically distributed and collaborative setting.

Today's architectural, urban design, product design and construction projects require close collaboration and teamwork among diverse specialists. Aggregating expertise is a primary goal for

all, in the increasing competition to successfully introduce new designs and products on a global scale. MIT is creating a network of new teaching and research design studios as prototypes for the next generation of design environments; physical walls will pose no barrier to reaching out to colleagues, clients and intellectual resources. In the design studio, students are able to:

- Model, visualize, and analyze design proposals using advanced geometric modeling and computer graphics technology
- Discuss projects with design critics and distant collaborators, both industrial and academic, via computer-integrated teleconferencing
- Access, via MIT's internal network and the Internet, extensive databases of images, digital models, and other reference material.
- Automatically convert digital geometric models into physical models and prototypes through use of computer-integrated fabrication techniques

### **Support to Date**

Sun Microsystems and IBM have become industry partners in the Design Studio of the Future project, contributing more than \$1.2M in equipment. Compaq Computer, Acer Computer, Fluor Daniel, PictureTel, Autodesk, and Lotus are also sponsors or partners in the DSOF. MIT has provided substantial physical and telecommunications infrastructure for the project, including a commitment of \$7M for new electronic studio space in the School of Architecture and Planning, and support for renovation of studio space in Civil Engineering. In addition, MIT continues as part of a \$3.5M inter-university collaborative (the Product Realization Consortium), funded by NSF and ARPA, to define and implement the collaborative electronic design-to-manufacturing process.

MIT is seeking collaborations with companies who share an interest in the future of electronically supported design work environments. Of particular interest are joint projects to develop support systems and test them on practical studio work, and to explore the educational uses of digital imaging and video teleconferencing. As most of the design studio prototype is now in place, the Design Studio of the Future project is recruiting selected industry sponsors and partners: companies with an interest in electronic integration of distributed sites and work environments, and in collaborative design (mechanical, civil engineering, electronic, or architectural). The program is making every effort to ensure that it addresses major issues facing industry in the next five to ten years.

The DSOF seeks to address major challenges facing design collaboration.

DSOF plans to:

- Involve multiple engineering, architecture, and planning disciplines to help address data representation, analysis, visualization and exchange issues
- Engage in projects involving multiple cultures to clearly identify multicultural conferencing issues
- Evaluate opportunities for new collaboration tools by designing real consumer and industrial

products

- Prototype new user-driven tools and interfaces more tailored to design needs
- Identify new applications for distance learning and collaboration
- Make extensive use of desktop conferencing systems to provide easy access for students to communicate with remote team members, reviewers, and design crits/consultants

## **Program Objectives**

The DSOF program is a collaboration of design and engineering faculty at MIT with three key objectives:

- Build a working prototype of a switched network of design studios, labs and specialized facilities, called StudioNet(TM), to simulate the networked engineering and design organization of the future
- Create an environment for collaborative design through coursework and industry-sponsored projects for experimentation with StudioNet
- Pursue focused research to push the frontiers of design and collaborative engineering environments

## **Program Components**

The DSOF program brings together design practice, education, and research in an interdisciplinary environment to accomplish its objectives

## **Faculty Leadership**

### **William J. Mitchell - Advisory Chair**

William J. Mitchell is Professor of Architecture and Media Arts and Sciences and Dean of the School of Architecture and Planning at the MIT. He teaches courses and conducts research in design theory, computer applications in architecture and urban design, and imaging and image synthesis. He consults extensively in the field of computer-aided design and was the co-founder of a California software company. His most recent book, *City of Bits: Space, Place, and the Infobahn* was published by the MIT Press in June of 1995. In addition to numerous articles, Mitchell is also the author of *The Reconfigured Eye: Visual Truth in the Post-Photographic Era* (MIT Press, 1992), which deals with the social and cultural impact of digitally altered photographs and synthesized photorealistic scenes; *Digital Design Media*, with Malcolm McCullough (Van Nostrand Reinhold, second ed., 1995; orig. publ. 1991); *The Logic of Architecture: Design, Computation, and*

Cognition (MIT Press, 1990); The Poetics of Gardens, with Charles Moore and William Turnbull (MIT Press, 1988); The Art of Computer Graphics Programming, with Robin S. Liggett and Thomas Kvan (Van Nostrand Reinhold, 1987); and Computer-Aided Architectural Design (Van Nostrand Reinhold, 1977). With Patrick Purcell and Malcolm McCullough he edited, and contributed essays to, The Electronic Design Studio (MIT Press, 1990).

### **Woodie C. Flowers - Faculty Chair**

Woodie Flowers is the Pappalardo Professor of Innovation in the Department of Mechanical Engineering and is Director of MIT's New Products Program. He received his S.M., M.E. and Ph.D. degrees from MIT. His current research includes work on micro-computer-controlled artificial legs, the creative design process, and computer-aided design systems. He has contributed to several new design courses and introduced MIT's well-known course "Introduction to Design." Professor Flowers also served as host for the PBS television series Scientific American Frontiers from 1990 to 1993, was elected to the National Academy of Engineering in 1994, and was honored in 1994 as a MacVicar Faculty Fellow for his extraordinary contributions to undergraduate education. His interests include design education, biomechanical engineering, electromechanical systems, computer-aided design systems, and product design.

### **John R. Williams**

John R. Williams is an associate professor of Civil and Environmental Engineering and director of the Intelligent Engineering Systems Laboratory at MIT where he specializes in computational design. He is a principal on MIT's ARPA-TRP Design Studio of the Future effort to develop computer and communications technology for the design and rapid prototyping of products. Before coming to MIT in 1990 he was Vice President of research at Applied Mechanics Inc. of Colorado and Managing Principal in Charge of the Advanced Technology Group at Dames and Moore, Inc. During this time he was responsible for the design and development of commercial analysis software. At MIT he specializes in the development of interactive simulation techniques for the design and rapid prototyping of products. He is presently working on developing fast physics for virtual reality based on wavelet and discrete element methods, and teaches object oriented software design methods.

## References

1. Watson, Robert., *Groupware in the 21<sup>st</sup> Century*. Praeger Publishers, CT. 1994.
2. Ellis, C.A., Gibbs, S.J., Rein, G.L., Groupware Some Issues and Experiences. *Commun. ACM* 34, 1 (Jan 1991), 384-402.
3. Briggs, Robert O., Nunamaker, Jay F., Getting a Grip on Groupware. In *Groupware in the 21<sup>st</sup> Century*. Praeger Publishers, CT. 1994. pp. 66.
4. Rooney, Paula., PC WEEK Online. April 12, 1996.
5. Taninecz, George., *Blue Sky Meets Blue Sky*, Industry Week, V. 244, No. 23, Dec 18, 1995. pp. 48-52.