

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

128677

MİMARİ YAZILIMLARIN GELİŞTİRİLMESİNDE
ARAYÜZ TASARIMI

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Mimar Emrah TÜRKYILMAZ

F.B.E Mimarlık Anabilim Dalı Bilgisayar Ortamında Mimarlık Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Oya PAKDİL

Prof. Dr. Ayfer Aytaç (YTD) / *Ayfer Aytaç*

Prof. Dr. Koray Göhan / *Koray Göhan* (İKÜ)

Prof. Dr. Oya Pakdil / *Oya Pakdil*

İSTANBUL, 2002

İÇİNDEKİLER

	Sayfa
KISALTIMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÖNSÖZ.....	vii
ÖZET.....	viii
ABSTRACT.....	ix
1. GİRİŞ.....	1
1.1 Araştırmanın Amacı.....	1
1.2 Araştırmanın Kapsamı.....	2
1.3 Araştırmanın Yöntemi.....	2
2. BİLGİSAYAR TEKNOLOJİSİ.....	3
2.1 Bilgisayarların Gelişme Süreci.....	3
2.1.1 Matematiksel Hesap ve Hesap Makinesi.....	3
2.1.2 19. Yüzyıl Ve Sanayi Devrimi.....	5
2.1.3 Makine Biliminin Ortaya Çıkışı ve Otomat Biliminin Gelişmesi.....	9
2.1.4 Felsefi ve Düşünsel Açından Tasarlanabilir Olma.....	10
2.1.5 Simgesel Mantık ve Matematik Bilimindeki İlerlemeler.....	11
2.1.6 Teknoloji Devrimi ile Teknik Olarak Üretilebilir Olma.....	13
2.1.7 Sanayi Toplumlarının Gelişmesi.....	14
2.1.8 II. Dünya Savaşı'nın Etkileri.....	15
2.1.9 Kayıtlı Program Kavramı.....	17
2.2 Modern Bilgisayarlar.....	18
2.2.1 İlk İşletim Sistemleri.....	20
2.3 Kişisel Bilgisayar Kavramı.....	21
2.3.1 Xerox Alto Araştırmaları.....	21
2.3.2 Microsoft Windows İşletim Sistemi.....	22
3. ARAYÜZ KAVRAMI.....	24
3.1 İnsan – Bilgisayar Etkileşimi (İBE) nedir?.....	24
3.2 Arayüz nedir?.....	24
3.2.1 Günlük Yaşamdan Arayüzler.....	26
3.2.2 Arayüz Tasarımının Diğer Disiplinlerle İlişkisi.....	26
3.2.3 Arayüz Tasarımının Önemi.....	27
3.2.4 Bilgisayar Biliminde Arayüz Tasarımının Önemi.....	27
3.2.5 İyi Bir Arayüz Tasarımı Nasıl Olmalıdır?.....	28
3.2.6 Arayüz Tasarımının Genel Prensipleri.....	28
3.2.7 Arayüz Tasarımında Tasarım Süreci.....	29
3.2.8 Kullanılabilirlik (Usability) nedir?.....	35
3.2.8.1 Kullanılabilirlik Tasarımı.....	36
3.2.8.2 Kullanılabilirlik Tasarımının İlkeleri.....	36

3.2.9	Arayüz Tasarımında Değerlendirme	40
3.2.9.1	Değerlendirme (Evaluation) Nedir ?	40
3.2.9.2	Değerlendirme Yöntemleri.....	40
4.	MİMARLIKTA BİLGİSAYAR KULLANIMI	43
4.1	Bilgisayar Destekli Tasarımın Gelişme Süreci	43
4.2	Mimari Tasarımda Bilgisayar Kullanımı	49
4.2.1	Simülasyon Aracı	50
4.2.2	Modelleme Aracı.....	52
4.2.2.1	Genel Modelleme Yazılımları.....	53
4.2.2.2	Görsel Modelleme Yazılımları.....	56
4.2.2.3	Mimari Modelleme Yazılımları.....	58
4.2.2.4	Taşıyıcı Sistem Tasarımı Yazılımları.....	62
4.2.2.5	İş Yönetimi Yazılımları.....	63
4.2.3	Ortak Çalışma (Collaboration)	63
4.2.3.1	Anında Haberleşme Yazılımları:.....	63
4.2.3.2	Yazı, Ses ve Görüntü ile Haberleşme Araçları:	64
4.2.3.3	Dosya Transferi ve Dosya Paylaşımı:	65
4.2.4	Sanal Tasarım Stüdyosu (Virtual Design Studio)	65
5.	BİR KAVRAMSAL MİMARİ YAZILIM ARAYÜZÜ ÖNERİSİ	67
5.1	İhtiyaç Listesi	67
5.2	İş Özeti	85
5.3	Kullanıcı Arayüz Önerisi.....	85
5.3.1	Çekme Menüler	86
5.3.1.1	File.....	87
5.3.1.2	Edit	88
5.3.1.3	Draw	89
5.3.1.4	Insert.....	90
5.3.1.5	Modify	91
5.3.1.6	Tools.....	92
5.3.1.7	Render	93
5.3.1.8	Design Studio	93
5.3.1.9	View	94
5.3.1.10	Help	95
5.3.2	Araç Çubukları	95
5.3.2.1	File.....	96
5.3.2.2	Transform	97
5.3.2.3	Edit	97
5.3.2.4	Text.....	98
5.3.2.5	Pen.....	98
5.3.2.6	Brush	99
5.3.2.7	Align.....	99
5.3.2.8	Arrange.....	99
5.3.2.9	Tools.....	100
5.3.3	Durum Çubuğu.....	101
5.3.4	Modify Araç Menüsü	102
5.3.4.1	Modify Pen	102
5.3.4.2	Modify Brush	102
5.3.4.3	Modify Text.....	103
6.	SONUÇ	105

KAYNAKLAR.....	107
INTERNET KAYNAKLARI.....	109
EKLER.....	112
Ek 1: i-cad.dpr Delphi 5 dosyası	113
Ek 2: main.pas Delphi 5 dosyası	114
Ek 3: about.pas Delphi 5 dosyası	129
Ek 4: i-CAD.iwz Installshield Express dosyası	131
Ek 5: i-CAD CD'si (i-CAD "kur" dosyası, yardım dosyaları, program kodları).....	134
ÖZGEÇMİŞ	135



KISALTMA LİSTESİ

ABC	Atanasoff Berry Computer
BDT	Bilgisayar Destekli Tasarım
CAAD	Computer Aided Architectural Design / Bilgisayar Destekli Mimari Tasarım
CADAM	Computer Augmented Drafting and Manufacturing
CHI	Computer-Human Interaction / Bilgisayar-İnsan Etkileşimi
DWG	AutoCAD Drawing File Format
DXF	Drawing Exchange File
EDVAC	Electronic Discrete Variable Automatic Computer
ENIAC	Electronic Numerical Integrator and Computer
GPL	General Public License / Genel Kamu Lisansı
GUI	Graphical User Interface
HCI	Human-Computer Interaction / İnsan-Bilgisayar Etkileşimi
HF	Human Factors / İnsan Faktörleri
HMI	Human-Machine Interface / İnsan-Makine Arayüzü
IBM	International Business Machine Company
IBM/PC	International Business Machine Company Personal Computer
İBE	İnsan-Bilgisayar Etkileşimi
MMI	Man-Machine Interface / İnsan-Makine Arayüzü
OMI	Operator-Machine Interface / Operatör-Makine Arayüzü
PC	Personal Computer / Kişisel Bilgisayar
SSEC	Selective Sequence Electronic Calculator
TTF	True Type Font
UCD	User-Centred Design / Kullanıcı Merkezli Tasarım
UID	User-Interface Design / Kullanıcı Arayüzü Tasarımı
WYSIWYG	What You See Is What You Get

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Pascal'ın icat ettiği hesap makinesi.....	5
Şekil 2.2 Babbage'ın Kalan Makinesi.....	6
Şekil 2.3 Babbage'ın Analitik Makinesi.....	7
Şekil 2.4 Hollerith'in Sınıflandırma Makinesi.....	9
Şekil 2.5 İlk bilgisayar ENIAC.....	16
Şekil 2.6 Modern bilgisayarların gelişimi.....	19
Şekil 3.1 Rideout (1991) tarafından geliştirilen kullanılabilirlik hedef çizgisi.....	30
Şekil 3.2 Orijinal tasarım konsepti.....	31
Şekil 3.3 Prototip çalışmasının iki boyutu.....	34
Şekil 3.4 Renk seçimi ile ilgili örnek ekran.....	35
Şekil 3.5 Öğrenme eğrisi.....	37
Şekil 3.6 Likert ve anlamsal değişkenlik skalaları.....	39
Şekil 4.1 Tasarım, modelleme ve analiz ilişkisi.....	52
Şekil 4.2 Autodesk AutoCAD 2000.....	54
Şekil 4.3 BricsNet IntelliCAD.....	55
Şekil 4.4 Bentley MicroStation.....	56
Şekil 4.5 Kinetix 3DStudioViz.....	57
Şekil 4.6 Cinema4D.....	58
Şekil 4.7 Autodesk Architectural Desktop.....	59
Şekil 4.8 Nemetschek Allplan.....	60
Şekil 4.9 Graphisoft ArchiCAD.....	61
Şekil 4.10 İdeYapı IdeCAD.....	62
Şekil 5.1 i-CAD ekranı.....	86
Şekil 5.2 Çekme menüler.....	87
Şekil 5.3 File menüsü.....	88
Şekil 5.4 Edit menüsü.....	89
Şekil 5.5 Draw menüsü.....	90
Şekil 5.6 Insert menüsü.....	91
Şekil 5.7 Modify menüsü.....	92
Şekil 5.8 Tools menüsü.....	93
Şekil 5.9 Render menüsü.....	93
Şekil 5.10 Design Studio menüsü.....	94
Şekil 5.12 View menüsü.....	94
Şekil 5.14 Help menüsü.....	95
Şekil 5.15 Araç çubukları.....	96
Şekil 5.17 File Araç çubuğu.....	96
Şekil 5.19 Transfrom araç çubuğu.....	97
Şekil 5.21 Edit Araç çubuğu.....	98
Şekil 5.23 Text araç menüsü.....	98
Şekil 5.25 Pen araç menüsü.....	98
Şekil 5.27 Brush araç menüsü.....	99
Şekil 5.29 Align araç menüsü.....	99
Şekil 5.31 Arrange araç menüsü.....	100
Şekil 5.33 Tools araç menüsü.....	101
Şekil 5.35 Durum çubuğu.....	101
Şekil 5.37 Modify Pen araç menüsü.....	102
Şekil 5.39 Modify Brush araç menüsü.....	103
Şekil 5.41 Modify Text araç menüsü.....	104

ÖNSÖZ

Çalışmalarım esnasında büyük sabır gösteren değerli tez danışmanım Prof. Dr. Oya Pakdil'e ve bana yardımlarını esirgemeyen eşim Arş. Gör. Çiğdem Canbay'a teşekkür ederim.



ÖZET

Günümüz teknolojisindeki hızlı gelişme, en belirgin olarak bilgisayar teknolojileri alanında görülmektedir. Diğer meslek alanları gibi mimarlık mesleği de bilgisayar teknolojisindeki gelişmelerden etkilenmektedir. Meslek eğitiminde de lisans aşamasından başlayarak bilgisayar kullanımı eğitimin bir parçası haline gelmiştir.

Mimari yazılımlar, bilgisayar teknolojisinin mimarlıktaki kullanım alanını oluşturmaktadır. Bu alandaki en büyük sorun, mimari yazılımların yeterli kullanılamamasıdır. Bu çalışmanın amacı, mimari yazılımların kullanım alanlarını inceledikten sonra, bir mimari yazılım arayüzü için gereksinimleri belirlemektir. Çalışmada, bilgisayar teknolojilerinin kullanımı, bilgisayar kavramından yola çıkılarak incelenmiştir. Bilgisayar destekli tasarım yazılımları hakkında araştırma yapılmıştır. Mimari yazılımların verimli olarak kullanılması için arayüz faktörünün önemi araştırılmıştır. Arayüzün tarihçesi, arayüz kavramı ve insan bilgisayar etkileşimi konuları incelenmiştir. Belirlenen konular ile ilgili literatür çalışması yapıldıktan sonra, bir mimari yazılımın ihtiyaç listesi belirlenmeye çalışılmıştır. Belirlenen ihtiyaç listesi ile kavramsal bir mimari yazılım arayüzü önerisi yapılmıştır.

Anahtar kelimeler: mimari yazılımlar, arayüz, bilgisayar teknolojileri, bilgisayar destekli tasarım

ABSTRACT

In view of today's ultimate technological leap is mostly seen on computer technologies. Just like other professions, architecture too, is under influence of this fast changing technological evolution. Due to this change, the use of computers has also become a part of the professional education of architecture.

The main use of computer technologies in the profession of Architecture is the architectural drafting software. According to the researcher, the main problem in the use of this software is the lack of productivity caused by the inadequate design of the program interfaces. The optimal solution for the problem of interface design for a more productive use of the architectural drafting software will be the main concern of this study.

Throughout this study, computer technologies, has been investigated over the concept of personal computers. The historical perspective of the software production, -architectural drafting software and other computer aided design software- has been investigated. The history of interfaces, the concept of interface, human-computer interactions, has been the main topics of research to find an answer to the question; what shall be the optimal interface design for a more productive use of the architectural drafting software?

According to the results of the research, the systematic of the use of the architectural drafting software, has been exposed. The properties of an optimal interface design for a more productive use of the architectural drafting software has been gathered and a connectional architectural drafting software proposition has been made.

Keywords: Architectural Software, User interface, Computer Technology, Computer Aided Design

1. GİRİŞ

Günümüzde teknolojideki hızlı gelişme, en belirgin olarak bilgisayar teknolojileri alanında görülmektedir. Bilgisayar teknolojisindeki gelişmeler, diğer meslek alanları gibi mimarlık mesleğini de etkilemektedir. Bilgisayar teknolojilerinin hem yazılım hem donanım olarak mimarlık mesleği ile ilişki içinde olduğunu görmekteyiz. Meslek eğitiminde de lisans aşamasından başlayarak bilgisayar kullanımı eğitimin bir parçası haline gelmiştir.

Lisans eğitimi aşamasından başlayarak, bilgisayar teknolojilerinin mimarlıkta verimsiz olarak kullanılması en büyük problemlerden biridir. Verimlilik iki yönden ele alınırsa;

- Mimarın yapması gerekli standart işlerde, zamanın iyi olarak kullanılamaması,
- Buna bağlı olarak düşünme sürecindeki yetersizlik, mimarın esas problem alanı olan tasarım işine yeterli zaman ayıramaması ve böylelikle tasarım fikrinin geliştirilememesi.

Mimarlıkta bilgisayar teknolojilerinin verimsiz olarak kullanılması probleminin altında mimari tasarımda yazılım kullanımının yetersizliği, mimari yazılımlarının kullanışsızlığı, mimari yazılımlarda teknolojilerin yeterince kullanılamaması, bazı mimarların yeni teknolojilere sıcak bakmaması gibi pek çok başka problemler de yer almaktadır.

Bütün bu problemleri ortaya çıkartan iki ana faktör bulunmaktadır:

- Bilgisayar teknolojilerinin yazılım üreten firmaların tekelinde olması,
- Bilgisayar teknolojilerinin kullanımında donanımlar ile sınırlı olmak.

Yazılım endüstrisinde, üretilen yazılımların belirli oranlarda satılması hedeflenir; bu oranlara ulaşıldıktan sonra yeni versiyonların üretilmesine başlanır. Dolayısıyla tekeli olarak adlandırabileceğimiz bu sistem yüzünden bir kerede en mükemmel ulaşmak mümkün olmamaktadır. Donanım konusunda da aynı mantık geçerlidir. Donanım teknolojisi, bilimsel sınırlar dahilinde, yeni teknolojiler üretmektedir. Bu teknolojik gelişmeler hedef kitleye ulaşmadan dolayısıyla belirli oranlarda satılmadan önce onları geliştirerek daha yeni teknolojileri üretmek mümkün olmamaktadır.

1.1 Araştırmanın Amacı

Bilgisayar teknolojisinin mimarlıktaki kullanım alanını mimari yazılımlar oluşturmaktadır. Bilgisayar teknolojilerinin mimarlık alanındaki en büyük sorunu, mimari yazılımların verimsiz olarak kullanılmasıdır. Bu çalışmanın amacı mimari yazılımların nasıl daha verimli kullanılabileceğini ve bunu sağlamada arayüz faktörünü incelemektir.

1.2 Araştırmanın Kapsamı

Bu çalışma kapsamında, öncelikle bilgisayar teknolojilerindeki gelişmeler kişisel bilgisayar (Personal Computer: PC) kavramından yola çıkılarak incelenmiştir. Mimari yazılımların nasıl daha verimli kullanılabilceđi sorusuna cevap aramak için yazılım dünyasının tarihçesi, mimari yazılımlar ve diđer BDT (Bilgisayar Destekli Tasarım) yazılımları hakkında araştırma yapılmıştır. Arayüzün tarihçesi, arayüz kavramı, insan bilgisayar etkileşimi konuları da mimari yazılımların verimli kullanılmasında arayüz faktörünün öneminin ne olduđu sorusuna cevap aramak için bu çalışmanın kapsamına dahil edilmiştir.

1.3 Araştırmanın Yöntemi

Bu çalışmada, belirlenen konular ile ilgili bir literatür araştırılması yapılmıştır. Bilgisayar kavramı araştırılarak bilgisayarın kullanım süreci irdelenmiştir. Daha sonra insan ve bilgisayar ilişkisi araştırılmış ve mimarlıkta bilgisayar kullanımı incelenmiştir. Son olarak, mimari yazılımların verimliliğini arttırmada kullanılabilcek bir arayüzün özelliklerinin neler olabileceđi tartışmak amacıyla kavramsal bir mimari yazılım arayüzü önerisi yapılmaya çalışılmıştır.

2. BİLGİSAYAR TEKNOLOJİSİ

Bu bölümde, bilgisayar teknolojisi alanındaki gelişmeler, tarihsel gelişim süreci içinde ele alınmıştır. Günümüzde mimarlık mesleğinin sürdürülebilirliği biçimini etkileyen bilgisayar teknolojisi, “bilgisayar” kavramından yola çıkılarak incelenmiştir.

2.1 Bilgisayarların Gelişme Süreci

Bilgisayarın icadında çok eski tarihlerden başlayan birçok olayın ve düşünce sisteminin etkisi olmuştur. Bilgisayarın icadının gelişme sürecini şu şekilde sıralayabiliriz: (Ifrah, 2000)

- Matematiksel hesap ve hesap makinesi.
- 19. yy. sanayi devrimi.
- Makine biliminin ortaya çıkışı ve otomat biliminin gelişmesi.
- Felsefi ve düşünsel açıdan tasarlanabilir olma.
- Simgesel mantık ile matematik bilimindeki ilerlemeler.
- Teknoloji devrimi ile teknik olarak üretilebilir olması.
- Sanayi toplumunun gelişmesi.
- II. Dünya Savaşının etkisi.
- Kayıtlı program kavramı.

2.1.1 Matematiksel Hesap ve Hesap Makinesi

İnsan düşüncesi kalıcı değildir. Bu yüzden insanlar bir toplumsal grup halinde yaşamaya başladıkları ilk andan itibaren, düşüncelerini görsel olarak betimleyip, kalıcı hale getirme gereksinimi duymuşlardır. Bu gereksinimlerini karşılayabilmek için yazıyı ve sayıları kullanmaya çalışmışlardır. Bu yüzden yazının tarihi ile yazılı sayılar tarihi arasında açık bir ilişki vardır (24) (Ifrah, 2000).

Sayısal gösterimlerin gelişmesiyle matematiksel hesap ortaya çıkmıştır. Hesap sözcüğü tarih boyunca oldukça fazla sayıda anlamlar kazanmıştır. Sonunda, çok genel biçimde, bir kümenin (sayılar, vektörler, büyüklükler, fonksiyonlar v. b) öğelerine, sonuç denen çok genel ve belirli bir matematiksel varlıklar sınıfını elde etmek üzere uygulanan, çok belirli kurallar ve yöntemler kümesinin kullanılmasının adı haline gelmiştir (28).

Başka bir deyişle, hesap adı verilen işlemler, sonuç adı verilen şeyi elde etmek için çok belirli aritmetik, matematik veya mantıksal yasalara göre yöntemli olarak düzenlenmiş karmaşık düşünsel işlemlerdir (24).

Yunan filozofu Platon, Philebos (Haz Üzerine) adlı yapıtında Sokrates ile diyalogu sırasında şunu söyler: “Hesap yapamıyorsan, geleceğin hazları üzerine de düşünemezsin. Ve yaşamın, bir insanın yaşamı değil, bir istiridyenin veya deniz salyangozunun yaşamı olur” (Ifrah, 2000).

Hesap yapmayı kolaylaştıracak bir aygıt icat etmek, belki de insanlığın en eski düşlerinden biri olmuştur. Aristoteles şöyle diyordu. “Her alet buyrukla veya bilerek kendi işini yapabilseydi, mekik kendi kendine dokusa, gitar kendi kendine çalsaydı, ustanın çırağa, efendinin de köleye ihtiyacı kalmazdı” (Ifrah, 2000).

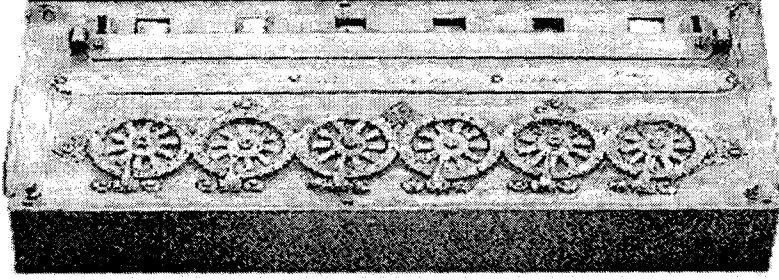
Hesap makineleri, insan hesabının eksikliklerini gidermeye yönelik, hesap yapmayı kolaylaştırma amacı taşıyan maddi aygıtlardır. Hesap makinesinin geleneksel anlamı, yalnızca sayılar ve büyüklükler üzerinde işlemler yapmaktır. Aritmetikçilerin yükünü hafifletmek için tasarlanmış olan bu aletler, doğrudan doğruya rakamlı betimlemeler üzerinde işlem yapmakta olup, belli sayıda ve görece işlemin ardından sonuç vermektedir (28).

İnsan hesabının ilk yardımcılarında biri, matematikçi John Napier tarafından 1617 ‘de icat edilen çarpım cetvelidir. Bu alet, birden dokuza kadar olan sayıların katlarını veren farklı sütunların, çarpılanın sırası ile yan yana konabileceği biçimde hareketli çubuklar üzerine kaydedildiği özel biçimli bir çarpım cetvelidir. Çarpmadaki ara çarpımlar, çarpanın rakamları karşısında yatay olarak okunur. Çarpmayı kolaylaştırmak için tasarlanmış olan bu güvenilir ve kesin dizgi 20. yy.ın başına dek Avrupa’da büyük bir başarı kazanmıştır (25).

20. yy.a kadar yapılan aletler gerçek hesap makineleri olamamışlardır. Sürekli olarak verileri yazmaya ve sonuçları okumaya kendini veren bir insan işlemcinin hesap işlemi içinde yer alması gerekmiştir. Dolayısıyla bu aletler rakamlar ile yapılan insan hesabı için yalnızca bir maddi yardımcı olmuşlardır (Ifrah, 2000).

Bu noktada, matematiksel hesap niye daha erken başlamadı sorusu akla gelebilir. Yunan matematikçileri sonsuz vidayı, dişlileri bildikleri ve kullandıklarına göre, bu farklı parçaları kullanarak bir hesap makinesi yapmalarını engelleyen nedir? Bunun nedeni, tekniğin yeterli olmamasıdır. Bunun dışında, Yunanlıların sayı sistemlerinde basamak ve “0” sayısı bulunmamaktadır (Ifrah, 2000).

Basit sayısal hesap makineleri ile ilgili arařtırmalarda Blaise Pascal'ın alıřmaları önemli bir dönüm noktasıdır. Pascal'ın icat ettiđi aritmetik hesap makinesi, bir dizi diřli arktan oluřmaktadır. Her bir arkın bir tam dönüřü, bir diđerine, onu bir derece ilerletecek řekilde aktarılmaktadır. Bu makine aslında tersine evrilemez bir düzeneđi bulunan bir toplama makinesidir ancak ıkarma yapma özelliđi de bulunmaktadır. Bu makinenin ilk otomatik hesap makinesi olduđu kabul edilmektedir (Edwards, 1996) (řekil 2.1).



řekil 2.1 Pascal'ın icat ettiđi hesap makinesi (47)

Pascal'ın alıřmaları ile birlikte, insanođlu, bıktırıcı ve uzun süren hesap işlemlerini yapmaktan kurtulmuřtur. Hesap makineleri, giderek genişleyen bir dizi düşünsel işlemin yerine getirilmesinde insanın yerini alamaya başlamıřtır. Hesap makineleri, onların işleyiřini düzenleyen fiziksel ve matematiksel yasaları bilmeyen kullanıcılara bile kolaylıkla hesap yapma olanađı sađlamaları ile yeni bir ađ açmıřlardır (Edwards,1996).

1673'te tasarlanmasına rađmen, ancak 1694'de yapılan Leibniz'in makinesi tüm temel aritmetik işlemlerini tamamen mekanik yollardan yapabilen ilk hesap makinesidir. Bu makinenin temelinde de Pascal'ın icat ettiđi hesap makinesinin özellikleri bulunmaktadır. Bu makinenin kullandıđı algoritma modern bilgisayarlarda da kullanılmaktadır (47).

2.1.2 19. Yüzyıl Ve Sanayi Devrimi

19.yy'da sanayinin ve sermayenin giderek artan önemi ve bunun sonucu olan kıyasıya rekabet, bankacılık ve saymanlık etkinliklerinde büyük bir gelişmeye yol açmıřtır. Bu da, bu işlere uyarlanmış donanımları kullanma konusunda gittikçe artan bir gereksinim doğurmuş ve böylece hesap makineleri sanayisinin yükseliřini beraberinde getirmiřtir (Ifrah,2000).

19. yüzyılda pek çok önemli gelişmeler olduğu görülmektedir. Bu gelişmeler kısaca gözden geçirilirse, ilk olarak otomatik dokuma tezgahından bahsetmek gereklidir. Otomatik dokuma tezgahı 1801'de Joseph Marie Jacquard tarafından icat edilmiştir. Bu alet, delikli kartlar aracılığıyla çalışmaktadır ve delikli kartlar dokunacak dokumaya desen vermeye yaramaktadır. Dokumaya desen vermek üzere, sert bir kartonun belirli yerlerine delikler açılması ile bir programlama dili oluşturulmuştur (Kurzweil,1992).

İlk sayısal hesap makineleri, "Kalan Makineleri" (Difference Engine) fikrinin gerçekleştirilebilir olduğu 19.yy.da kanıtlanmıştır. Kalan makineleri, bir kumanda düzeneği ile önceden belirlenmiş bir sürece göre, otomatik olarak zincirleme işlemler yapabilen ilk sayısal hesap makineleridir. Bu makinelerin böyle adlandırılmalarının nedeni, belirli bir başlangıç değeri üzerine, ardı ardına toplamalar yaparak matematiksel hesap yapmalarıdır. Bu makineler basamak değeri ile hesap yapamamaktadırlar (6) (Şekil 2.2).

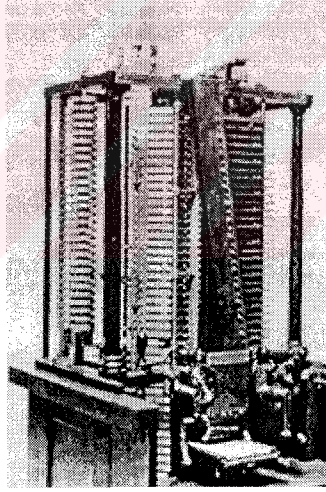


Şekil 2.2 Babbage'ın Kalan Makinesi (12)

Bu makinenin fikri, ilk kez 1786'da Alman askeri mühendisi H. Muller tarafından ortaya atılmıştır. Bu fikrin gerçekleştirilebilir olduğu ise 1822'de İngiliz matematikçi Charles Babbage tarafından kanıtlanmıştır. Bu tasarı hiçbir zaman tam anlamıyla gerçekleştirilmese de bilgisayarın icadının gelişme sürecinde en önemli dönüm noktalarından biri olmuştur (Lindgren, 1996).

Charles Babbage'a göre, makinelerin aynı hareketi defalarca yineleyebilmesi aynı adımlarla tekrarlanan matematiksel işlemler ile uyum içindedir (Kurzweil,1998). Babbage, buhar ile çalışan ve denklem çözebilen ilk hesap makinesi olan 'Kalan Makinesi' (Difference Engine) için gerekli maddi desteği İngiliz hükümetinden sağlamıştır. Ancak üzerinde on yıldan fazla çalışma yapmasına rağmen, bu tasarısını tam anlamıyla gerçekleştirememiştir (Computers Society). Fikir gerçekte ilk kez 1853 yılında, Babbage'ın çalışmalarından esinlenen George ve Edward Scheutz'un ülkelerinin maddi desteği ile güvenilir şekilde işlem yapan ve sonuçları basma aygıtı bulunan bir kalan makinesi yapmaları sayesinde gerçekleştirilmiştir (Lindgren, 1996).

Babbage, "Kalan Makinesi" üzerinde çalışırken, icadının özel amaçlı bir makine olduğunu ve sadece belirli işlemleri yapabildiğini fark etmiştir. Böylece 1853'de "Analitik Makine" (Analytical Engine) adını verdiği yeni bir projeye başlamıştır. Analitik Makine, modern bilgisayarların temel parçalarından oluşmaktadır (12). Analitik Makine hiçbir zaman çalışmamış olmasına rağmen, büyük hesap dizilerini ve mantık problemlerini çözecek şekilde programlanabilmektedir (Lindgren, 1996) (Şekil 2.3).



Şekil 2.3 Babbage'ın Analitik Makinesi (35)

Bütün bu gelişmeler yaşanırken, İngiliz Astronomi Birliği (British Astronomical Society) ilk altın madalyasını "Matematiksel Tabloları Hesaplamak için Kullanılan Makine Uygulamalarının Gözlemleri" (Observation on the Application of Machinery to the Computation of Mathematical Tables) adlı bildirisi için Charles Babbage'a vermeyi uygun bulmuştur (Kurzweil, 1992).

Babbage'ın yardımcısı olan Ada Augusta King, 1843'de L.P. Menabrea'nın Analitik Makine için yazdıklarını İngilizce'ye çevirdikten sonra, kendi notlarını da ekleyerek tekrara yayımlamıştır. Dünyanın ilk bilgisayar programcısı olarak kabul edilen Ada King, insan zekasının makineler tarafından taklit edilebileceği konusunda kuramlar geliştirmiştir (Kurzweil, 1992).

Babbage , “Kalan Makinesi” üzerinde yaptığı çalışmaların sonucunda 1847'de 21 adet imalat çizimini tamamlamıştır. Babbage'ın ölümünden sonra oğlu, Kalan Makinesi'nin temel aritmetik işlemleri yapan kısmını imal ederek çeşitli üniversitelere yollamıştır. Kalan Makinesi 1991 yılında, Kensington Bilim Müzesi tarafından aslına uygun bir şekilde imal edilmiş ve basit bazı hatalarının dışında mükemmel çalıştığı görülmüştür (12).

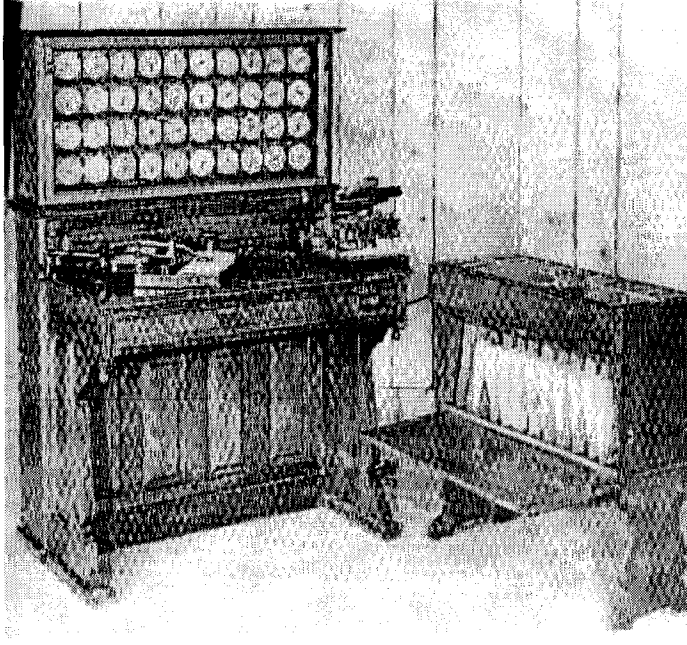
1847'de George Boole, simgesel mantık ve aritmetik üzerine kurduğu teorileri yayımlamıştır. İkilik sistem üzerine kurduğu mantık ve aritmetik, modern bilgisayarların temelini oluşturmaktadır (Kurzweil, 1992).

1877'de William Thompson, matematiksel problemlerin çözümü için makinelerin programlanabileceğini ispat etmiştir. 1883'de ise Thomas Edison elektrik iletimini olanaklı olduğunu ispatlamıştır. Böylece elektrik, yeni bir teknoloji olarak her alanda kullanılmaya başlanmıştır (47).

19.yy.'ın sonlarına gelindiğinde, Endüstri Devrimi gelişerek tüm dünyaya yayılmaya devam etmektedir. Sayıların üzerine kurulu olan icatlar daha karmaşık olmaya başlamıştır ve mühendislik için gerekli olan hesaplamaları hızlandırmak için hassas çalışan makinelere ihtiyaç duyulmaya başlanmıştır. 1890 yılına gelindiğinde, birçok firma, elektrikli hesap makinesi üretimine geçmiştir (28).

1890 yılında, Amerikan Sayım Bürosu, gittikçe daha çok kişi hakkındaki istatistiksel bilgiler kümesini, kısa bir zaman içinde çözümleme sorunu ile karşı karşıya kalmıştır. Bu yüzden, yeni sayım için kesin, güvenilir ve hızlı bir istatistiksel sistem geliştirebilecek bir mucit bulmak amacıyla bir yarışma düzenlemiştir. Yarışmayı, Hermann Hollerith “Sınıflandırma Makinesi” (Tabulated Machine) ile kazanmıştır. Bu makine, yedi yıl sürmesi planlanan veri işleme sürecini, altı haftaya indirmeyi başarmıştır (12).

Sınıflandırma Makinesi, sert karton üzerindeki delikler sayesinde birimleri sayabilen elektromekanik bir istatistik makinesidir. Bu makinede her bir delikten elektrik akımı geçmekte ve toplayıcının ibresini bir derece iletirmektedir. Hollerith, bu makinede Jacquard'ın geliştirdiği delikli kart sistemini kullanmıştır (Kurzweil, 1992) (Şekil 2.4).



Şekil 2.4 Hollerith'in Sınıflandırma Makinesi (8)

Hollerith'in makinesinin "Çizelgeleyici" adıyla bilinen mekanografik makinelerin gerçek atası olduğunu söylemek mümkündür. Böyle adlandırılmalarının nedeni, sonuçları çizelgeler halinde vermeleridir. Hollerith, yarışmayı kazandıktan sonra, 1896'da daha sonra IBM adını alacak Tabulating Machine Company'yi kurmuştur (Kurzweil, 1992).

2.1.3 Makine Biliminin Ortaya Çıkışı ve Otomat Biliminin Gelişmesi

Tek başına, yani insanın hiçbir müdahalesi olmadan, önceden belirlenmiş bir süreci izleyerek zincirleme işlem yapabilen bir makineye duyulan ihtiyaç makine biliminin ortaya çıkmasına neden olmuştur. Makine biliminin bu ihtiyacı karşılamak üzere çalışması sonunda bilgisayarların icadına giden yol açılmıştır (Ifrah, 2000).

Otomat bilimi ise, makine bilimine yardımcı bir bilim olarak gelişmiştir. Otomatlar, kendi hareketleriyle kendiliğinden tepki gösterme eğilimi bulunan aygıtlardır ve Eskiçağ'dan beri geliştirilmektedirler (Ifrah, 2000).

Başlangıçta, yapay otomatlar yaşam yanılması vermeye yönelik hayvan veya insan görünümlü aygıtlardır. Dolayısıyla bunlar bilgi verme otomatları değildirler. Zamanı ve gökbilimsel verileri tam olarak ölçme zorunluluğundan ve yapay otomatları dünya üzerine bilgi vermeye yönelik bir aygıt haline getiren kökten bir kavram değişikliği ilk defa Yunanistan ve Çin’de, daha sonra İslam Dünyasında ve Batı Avrupa’da gerçekleştirilmiştir. Bu değişikliklerin sonucu olarak, yapısı bakımından dünya hakkında bilgi vermek üzere tasarlanmış ilk otomatlar, yani saatler, icat edilmiştir (47).

Otomatları programlayabilmek için, insan ile makine arasında, önceden belirlenmiş bir iletişim aracı tasarlamak gerekli olmuştur. Otomat, bu kumanda aygıtının yönlendirmesi ile çalışmaktadır ve dolayısıyla verilen komutların çok kesin bir kurallar kümesine karşılık geldiği bir makine yaratmak bir zorunluluk olarak ortaya çıkmıştır. Verilen komutların belirli kurallar kümesine karşılık gelmesi fikri, bilgisayarların temel fikirlerinden birini oluşturmuştur (Ifrah, 2000).

2.1.4 Felsefi ve Düşünsel Açıdan Tasarlanabilir Olma

Felsefi ve düşünsel açıdan bilgisayarın icadının gelişme sürecine etkileyen en önemli fikirler Charles Babbage tarafından ortaya atılmıştır. Babbage’ın Çözümleyici/Analitik Makinesi (Analytical Engine) bu anlamda bilgisayarın gerçek atasıdır (47).

Maurice d’Ocagne Analitik Makineyi şöyle tanımlar: “Bir sayı üzerinde herhangi bir aritmetik işlemler dizisini yapmaya ve yapılan tüm işlemler dizisinin sonucunu cebir imleri ile vermeye yönelik makinedir” (Ifrah,2000). 1834’den 1836’ya kadar kuramsal olarak tasarlanan Babbage’ın makinesi yapısı bakımından 50 bin rakamlı 1000 sayı üzerinde, aritmetik veya cebirsel işlemleri otomatik olarak çözmek için yapılmıştır. Çözümleyici makinenin dili Jacquard’ın geliştirdiği kart delme dizgesine dayanmaktadır (Kurzweil, 1998).

Çözümleyici makine kesinlikle kendi kendine bir şey yaratma iddiasında olmamıştır. Çözümsel bağıntıları veya gerçekleri tasarlama yeteneği bulunmadığından, zaten yapmayı bildiğimiz bir şeyi yapmamıza yardım eden bir araç olmuştur (6).

Babbage’ın makinesi şu parçalardan oluşmaktadır:

- Bir giriş ve çıkış aygıtı (input/output).
- Kumanda organı: bu organ sayıları doğru sıraya sokmak için bir aygıttan ötekine sayıların aktarımını yapmaktadır.

- Yapılan hesapların ara ve son sonuçlarını saklamaya yarayan bir bellek dizgesi (store): bu, makinenin hesaba girecek sayıları alıp hesap sonucu olarak elde edilenleri biriktirmeyi amaçlayan belleğidir.
- Makineye verilen sayılar üzerinde işlemler yapmakla görevli bir aygıt olan değirmen (mill): bu, makinenin aritmetik birimidir; burada sayılar istenen kurallara göre mekanik olarak birleştirilmektedir, başka bir deyişle bu, söz konusu hesabın sonuçlarını elde etmek için, dönüştürmek üzere makineye verilen verileri kullanarak hesapları yapmakla görevli iletişim organıdır.
- Sonuçları basmaya yarayan bir baskı düzeneği (printing device) (6).

Babbage'ın çözümleyici makinesi, işlem akışında hiçbir insan müdahalesi gerektirmediğinden, otomatik sayısal hesap makinesi kavramını, otomat kavramı ile birleştirmiştir (Ifrah, 2000).

Babbage'ın çözümleyici makinesi, kağıt üzerinde bile olsa, bugünkü bilgisayarların gerçek öncüsünü oluşturmuştur. Yıllar boyunca sürdürülen tüm araştırmalar bir şekilde mutlaka Charles Babbage'ın çalışmalarından etkilenmişlerdir (Ifrah, 2000).

Babbage'ın uzun yıllar süren araştırmalarına rağmen bilgisayarı keşfetmesini engelleyen dört nedeni vardır:

- Gelişmekte olmasına karşın, Babbage'ın çağında henüz onun kavramlarının düzeyine gelmemiş olan teknik donanımın kullanımına ilişkin bilginin eksik olması,
- Babbage'ın makinesinin tanımını matematiksel yapısından ayıramamış olması; başka bir deyişle, makinenin fiziksel donanımı yapısal incelemeden ayırmaya yönelmemiş olması,
- Matematik ile mantığın onun çağında belli bir sorunlar sınıfının çözüm algoritmasını simgesel olarak dile getiren program kavramını tasarlayacak ölçüde gelişmiş olmaması,
- Teknik açıdan, çözümleyici makinenin belleğinin çok sınırlı olması, dolayısıyla belli bir programın bütün yönergelerinin fiziksel olarak kaydedilememesi (6).

2.1.5 Simgesel Mantık ve Matematik Bilimindeki İlerlemeler

Babbage'ın çok önemli araştırmalarını karşın, onun yapıtı olan Analitik Makine 1920'lerin başında tam olarak gerçekleştirilmiş değildir (Ifrah, 2000).

Böyle bir hesap makinesi gereksiniminin kendini hissettirmesi için İkinci Dünya Savaşını beklemek gerekmiştir. İşte o zaman, simgesel mantık ile matematiğin çok önemli gelişmelerinin eşlik ettiği teknolojik ilerlemeyle birlikte, Babbage'ın düşü sonunda gerçek olmuştur (Ifrah, 2000).

1930-1940 yılları boyunca, makineyle büyük hesaplar yapma gereksinimi mutlak bir zorunluluk olarak kendini göstermiştir. Başka bir deyişle, klasik sayısal hesap makinelerinin çok açık yetersizliğinden ve analogik yöntemin yapısal sınırlılığından dolayı, çok daha gelişmiş makineler kullanma gereksinimi iki savaş arasında gittikçe artmıştır. Ve 1930'lu yıllardan itibaren yüz yıl önce Charles Babbage'ın çizdiği yol üzerinde kimi araştırmacılar gerçek bir çözümleyici hesap makinesi yapma çabasına girmişlerdir (24).

Çözümleme/Çözme/Analiz, Yunaca "analysis" kelimesinden gelir. Etimolojik bakımdan analiz, çözme, bir bütünü parçalara ayırma, hatta parçaları da daha küçük öğelere bölme eylemidir. Bu, matematikte kullanılan "ortaya konan bir problemi bir başkasına, onu da bir üçüncüye götüren, çözümü bilinen bir probleme varıncaya kadar böyle devam edilen" çözümleyici yöntemden başka bir şey değildir. Condillac'ın dediği gibi, "Çözümlemek, öğeler ayırmak, bağıntıları karşılaştırmak ve kavramaktır...". Dolayısıyla çözümleme bir nesnenin tam olarak öğelere ayrılması olarak tanımlanabilir (Ifrah, 2000).

Öyleyse bu yöntem, çözümünü bildiğimiz bir problemde yola çıkarak bir sonraki problemin çözümünü ondan türeterek devam ettiğimiz birleştirici yöntemle tam bir karşılık içindedir. Çözümleme yöntemi:

- Çözülebilir türden bir problemin çözümü için gerekli matematiksel kurallar ve işlemler bütününe belirlemek üzere o problemi bir başkasına götüren ve böyle devam eden bir ön araştırmayla başlayan;
- Amacı bu kuralları ve işlemleri gelişmiş bir makinedeki bir yönergeler programı biçiminde kullanmak olan;
- Söz konusu problemin, yönergeleri otomatik olarak ve belirlenmiş işlem sırası içinde uygulayan makine tarafından çözümlenmesiyle sona eren bir etkinliktir (24).

Otomat terimleriyle dile getirirsek, bu tip etkinliğe dayanan hesap makineleri için şu genel tanım verilebilir:

- Çözümleyici bir hesap makinesi, sıralı ve programlanabilir bir sayısal otomattır. Bu otomat esnek bir programlama dizgesi tarafından yönetilir.

Makinenin iç aygıtlarının maddi yapısından bağımsız bir değişebilir işletim organı vardır; program yönergelerinin işleyişi sırasında hiçbir insan müdahalesi gerekmez (47).

2.1.6 Teknoloji Devrimi ile Teknik Olarak Üretilbilir Olma

Elektronik teknolojisinin tarihi 1883'te, Amerikalı mucit Thomas E. Edison'un, boş bir tüpün içinde sıcak bir elektrottan soğuk bir elektrota akım geçebildiğini (tersi mümkün değildir) ve ince bir metal tel üzerine akım gönderildiğinde telin ısınarak serbest elektronlar yaydığını keşfetmesiyle başlamıştır (Kurzweil, 1998).

1904'te Edison'un "termoionik" deneyinin etkisinden yararlanmayı düşünen İngiliz fizikçi A. Fleming, bu yolla elektrik salınımlarını ortaya çıkarmıştır. Bu olayda, ısıtılan telin yaydığı serbest elektronlar levha üzerine doluşarak bir elektrik akımı yaratmaktadırlar. Böylece sonradan "Diyot" adıyla anılan "Salınım Valfi" veya "Flemming Valfi" icat edilmiştir (47).

Ardından, Amerikalı Lee De Forest, tüpün içine, levha ile tel arasına üçüncü bir elektrot eklemeyi düşünmüş ve 1907'de "Triyotu" keşfetmiştir. Böylece daha güçlü elektrik akımları elde edilmeye başlanmıştır (12).

Bütün bu çalışmalar, elektronik tüplerin gelişimini beraberinde getirmiştir ve radyo alıcılarının yapılmaya başlanmasıyla, radyofonik sanayi gelişmiştir (Ifrah, 2000).

Elektronik tüplerle yapılan çalışmalar ilerledikçe, bu boş tüplerin elektrik sinyallerinin yer değiştirmesi için kullanılabilmesi anlaşılmaya başlanmıştır. Böylece, elektronik mıknaatısları kullanmak yerine, aritmetik hesapları elektronik tüplerle yapmanın olanaklı olduğu keşfedilmiş ve bu ilerleme elektronik hesabın gelişmesine yol açmıştır. Bir bilgi kodlama düzeneği geliştirilmiştir ve elektronik hesap makineleri üzerinde çalışılmaya başlanmıştır (12).

Bilgisayarın icadının gelişme sürecinde, bilgisayarları teknik olarak üretilebilir duruma getiren en önemli aşamalardan biri "flip-flop" devresinin icat edilmesi olmuştur. İngiliz bilim adamları W. H. Eccles ve F. W. Jordan tarafından keşfedilen flip-flop, elektronik bir çift kararlı ikili devredir. 1919'da yapılan bu keşif, ilk ikili elektronik devrelerin yapılmasını sağlamıştır (47).

Elektronik, yapay hesap alanında, yüzyıllardır ortaya konmuş temel veya karmaşık her türlü soruna basit, çabuk ve çok etkili çözümler getirmekle kalmamış, çözümleyici hesap makineleri ile sıradan hesap makinelerinin çok sınırlı olan olanaklarını da aşmıştır. Bu teknoloji, çok çeşitli olanaklı kullanımlarıyla, insanlığa çok geniş ufuklar açan ve akla gelmeyecek veya uzun süre çözülemez diye görülmüş sorunların çözümünün günlük bir gerçeklik haline gelmesini sağlayan yeni kavramların gelişmesine yol açmıştır (Ifrah, 2000).

2.1.7 Sanayi Toplularının Gelişmesi

1930'lu yıllara gelindiğinde, özellikle Almanya ve Amerika'da hızlı gelişmelerin olduğu görülmektedir . Bu ülkelerden bazı bilim adamları pek çok teknolojik buluşa imzalarını atmışlardır.

1930'da Amerikalı bilim adamı Vannevar Bush "Diferansiyel Çözümleyici" (Differential Analyzer) adında bir makine icat etmiştir. Diferansiyel Çözümleyici 1930 yılının en önemli teknolojik ilerlemesidir (Rheingold, 1985).

Alman bilim adamı Konrad Zuse ise 1935'den 1939'a dek çözümleyici, mekanik ve ikili hesap makineleri tasarlayarak yapmaya çalışmıştır. Araştırmaları, cebirsel denklemlerin çözümünü amaçlayan Z1 ve Z2 makineleri üzerinde yoğunlaşmıştır. Zuse, 1935 yılında ailesinin oturma odasında Z1 adını verdiği bilgisayarı icat etmiştir. İkili sistemde çalışan bu makine anahtarlama sistemi ile çalışmaktadır (47) (12).

1937'de Alan Turing "Evrensel Makine" (Universal Engine) adını verdiği kavramsal bir fikir geliştirmiştir. Evrensel Makineye göre tanımlanabilen her algoritma hesaplanabilir nitelikte olmalıdır. Aritmetik işlemlerin hesabı üzerine çalışan diğer bilim adamlarından farklı olarak, Turing'in fikirleri sembol işleme üzerine kuruludur (1).

Turing, geliştirdiği kavramsal fikirden yola çıkarak "Turing Makinesi" (Turing Machine) adını verdiği teorik bir bilgisayar geliştirmiştir. "Hesaplanabilir Sayılar Hakkında" (On Computing Numbers) adlı makalesinde yayımladığı fikirleri Bertrand Russell ve Charles Babbage'ın fikirleri üzerine kuruludur. Alan Turing, Alonzo Church ile birlikte Church-Turing Tezini geliştirmiştir. Bilim adamları bu tezlerinde, insanın çözebildiği her problemin bir dizi algoritmaya indirgenebileceğini savunmaktadırlar. Bu düşünce aynı zamanda makine zekası ve insan zekası işbirliği fikrini ortaya atmıştır (1).

Bu arada çalışmalarına devam eden Zuse , 1938'de Helmuth Schreyer ile birlikte Z1'in devamı olan Z2 adlı makineyi icat etmiştir (Computer Society / Events in the history of computer) Ne var ki savaş öncesi icat edilen Z1 ve Z2 Zuse'nin çalışmalarını yaptığı Almanya dışında hiçbir zaman tanınmamıştır. Dolayısıyla Zuse'nin çalışmaları dünya ve gelişmekte olan bilgisayar endüstrisi üzerinde etkili olamamıştır (47) (12).

Amerika'da aynı yöndeki ilk önemli girişimler, Bell Telefon Laboratuvarlarına bağlı olarak çalışan fizikçi ve mühendis George Stibitz tarafından gerçekleştirilmiştir. Stibitz 1939'da kompleks sayı kümeleri üzerinde hesap yapabilen bir alet icat etmiştir. "Kompleks Sayı Hesaplayıcısı" (Complex Number Calculator) adı verilen bu makine, geniş ölçekli elektromanyetik anahtarlı bir hesap makinesidir. Bir yıl sonra, bu makine, telefon hatları üzerinden uzaktan kullanılan ilk hesaplayıcı olmuştur (12). Bu makine dünyada bilinen ilk gerçek elektronik hesap makinesidir (Ifrah, 2000).

Yine Amerika'da 1939'da John Vincent Atanasoff, Clifford Berry ile birlikte küçük bir ikili sayısal hesap makinesi geliştirmiştir. "ABC" (Atanasoff Berry Computer) adı verilen bu makine, doğrusal denklemleri çözmek için geliştirilen özel amaçlı bir makinedir ve ilk elektronik programlanamayan bilgisayar olarak kabul edilmektedir (6). Bu makine doğrusal denklem dizgilerinin çözümü için tasarlanmıştır. Modern bilgisayarların genel kavramlarını oluşturan elektronik aritmetik birimi ve tekrar üretilebilir hafıza ünitelerinden oluşmaktadır (16). Ne var ki bu makine fiziksel ve mantıksal yapısı bakımından başlangıç düzeyinde kalmıştır ve hiçbir zaman doğru olarak çalışmamıştır (Ifrah, 2000).

2.1.8 II. Dünya Savaşı'nın Etkileri

Çözümleyici hesabın gelişmesi asıl yükselişini 1940'larda, savaşın getirdiği bilimsel ve askeri gereksinimlerden ötürü, savaş sırasında göstermiştir. Bunun nedenleri şöyle sıralanabilir:

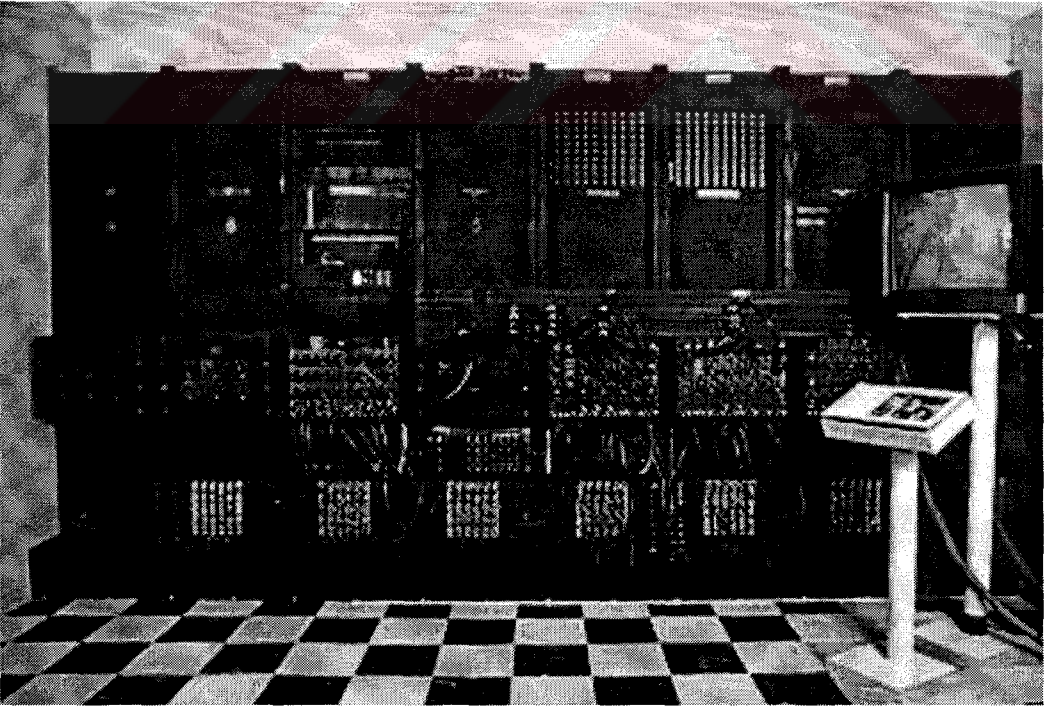
- Almanlar ile Japonların gizli iletişimindeki şifreli yazıyı en kısa sürede çözmeyi başarmanın çok önemli olması,
- Topçulukta daha etkili bir nişanlama için analogik hesap makinelerinininkinden daha fazla kesinlikte hesaplara ulaşma kaygısı,
- Simülasyon sorunlarını çözmeye gittikçe daha gerekli hale gelmesi,
- Hava akınlarında radar ile ilgili karmaşık hale gelen sorunları çözmeye gereksinimi (54).

Bu nedenlerin itici etkisi ile, dünyanın ilk çalışan bilgisayarı İngiltere'de Savaş Bürosu için çalışan "Ultra Grubu" tarafından yapılmıştır. "Robinson" adı verilen bu makine Almanların "Enigma" adlı şifreleme yöntemini çözmek için kullanılmıştır (Kurzweil, 1998).

Savaş sırasında çalışmalarına devam eden Zuse, 1941’de dünyanın ilk tamamen programlanabilir dijital bilgisayarı olan Z3’ü geliştirmiştir. Z3, Arnold Fast adında bir matematikçi tarafından programlanmıştır. Fast, çalışan bir bilgisayarı ilk kez programlayan bilgisayar programcısı olmuştur (40).

Alan Turing ve Ultra Grubu II. Dünya Savaşı sırasında Almanların şifreleme yöntemi olan Enigma’yı çözmek için bir dizi makine geliştirmiştir. Bu makinelerin sonuncusu olan “Colossus” 1943 yılında, savaşın sona ermesine kısa bir süre kala yapılmıştır. Bir önceki model olan Robinson’dan on kat daha hızlı çalışan Colossus, elektronik tüpler kullanılmaktadır (12). Bu makinenin tasarlanılmasındaki amaç, Naziler’in gösterdiği hızlı ilerlemeyi yakalamak olmuştur. Bu makine ikili mantıksal hesap yapabilmekte ve işlem sonuçlarını otomatik olarak yazı makineleri ile yazabilmektedir. Bu makinenin varlığı savaş sonuna dek gizli tutulmuştur (52).

1943 yılında, modern bilgisayarların atası sayılan “ENIAC’ın” (Electronic Numerical Integrator and Computer) üretim çalışmalarına başlanmıştır (12). Bu makine 60 ton ağırlığında ve 72m² yer kaplayan devasa bir yapıya sahiptir. 6 metre genişliğinde ve 12 metre uzunluğunda bir alanı kaplayan bu makine, bir çok düğmeden oluşan bir pano ile programlanabilmektedir (6) (Şekil 2.5).



Şekil 2.5 İlk bilgisayar ENIAC (Allison, 1996)

ENIAC'ın karmaşık bir işletim sistemi bulunmaktadır. İşletim yönergelerinin her biri kullanıcı tarafından el ile sayısal olarak girilmektedir. Bu yöntem hata oranını arttırmaktadır. Makine çok sık arıza yaptığından güvenilirliği azdır. ENIAC'ı icat eden bilim adamlarına göre makine %20 oranında doğru sonuç verebilmektedir (2).

ENIAC, üretildiği yıllarda, eşi görülmemiş bir hızda çalışmaktadır. Bir toplama 200 mikro saniyede, bir çarpma 2, 8 milisaniyede yapılabilmektedir (2).

1944 yılında IBM ve Harvard Üniversitesi, "Harvard Mark I" adında bir bilgisayar geliştirmiştir. "Mark I" Amerikalıların geliştirdiği ilk bilgisayar olup, genel amaçlı çalışan, geniş ölçekli ve elektromanyetik bir hesaplama makinesidir. Babbage'ın Fark Makinesi'nden esinlenerek yapılmıştır. Delikli kağıt bantlar ve vakum tüpleri ile çalışan Mark I, Amerikan Deniz Kuvvetleri için matematiksel denklemleri çözmeye kullanılmak üzere üretilmiştir (12).

Mark I, ENIAC'a göre çok daha yavaş çalışmaktadır. Bir toplama işlemi 0, 3 saniye ve çarpma işlemi 6 saniye sürmektedir. ENIAC'a göre 1000 kat daha yavaş olmasına karşın, Mark I çok daha güvenilir bir makine olmuştur (Ifrah, 2000).

2.1.9 Kayıtlı Program Kavramı

1940'lı yılların ortasından itibaren kayıtlı program kavramı geliştirilmeye başlanmıştır. John Von Neumann, 1945'te bilgisayarlar için program depolama kavramı ile ilgili bir teori yayımlamıştır (Kurzweil, 1998). Grace Murray Hopper, Mark I makinesi ile çalışırken, bulunduğu bir hatayı (bug) düzeltmek için "Hata Ayıklama Sistemini" (Debug) geliştirmiştir (12).

1947'de, IBM, "SSEC" (Selective Sequence Electronic Calculator) adı verilen bir makine geliştirmiştir. Bu makine, bir insan işlemcinin karar vermesine gerek olmadan, işlem sırasını kendi kendine değiştirmek için belirli bir program yönergelerini otomatik olarak kullanabilmektedir (28).

Bütün üstünlüklerine ve bütün başarılarına karşın, programlanmasındaki güçlüklerden dolayı SSEC yerine ENIAC veya Mark I kullanımı tercih edilmiştir. Bunun nedeni IBM SSEC, kayıtlı program kullanmayıp, Charles Babbage'ın fikirlerine dayalı olarak çalışmasıdır. SSEC kayıtlı program kullanmadığı için bilgisayar olarak sınıflandırılmamaktadır (Ifrah, 2000). SSEC, Babbage'ın Analitik Makinesi'nin güçlü bir versiyonu olabilmiştir (28).

SSEC'den sonra bütün bilgisayarlar Neumann'ın kayıtlı program teorisine göre geliştirilmeye başlanmış ve kayıtlı program teorisi gelecek kuşak bilgisayarların (modern bilgisayarlar) mimarilerini oluşturmuştur (27).

2.2 Modern Bilgisayarlar

Modern bilgisayarlar, yani programlanabilen bilgisayarlar ile ilgili ilk çalışmalar, John von Neumann tarafından başlatılmıştır. 1945 yılında, Neumann tarafından ortaya konulan kayıtlı program teorisi, modern bilgisayarların oluşumunu önemli ölçüde etkilemiştir. Bilgisayarların çalışma prensiplerini kolaylaştıran ve işlem kapasitelerini esnekleştirerek arttıran bu teoriden yola çıkan Eckert ve Mauckley, 1947 yılında "EDVAC" adlı bilgisayarı geliştirmişlerdir (Kurzweil, 1998).

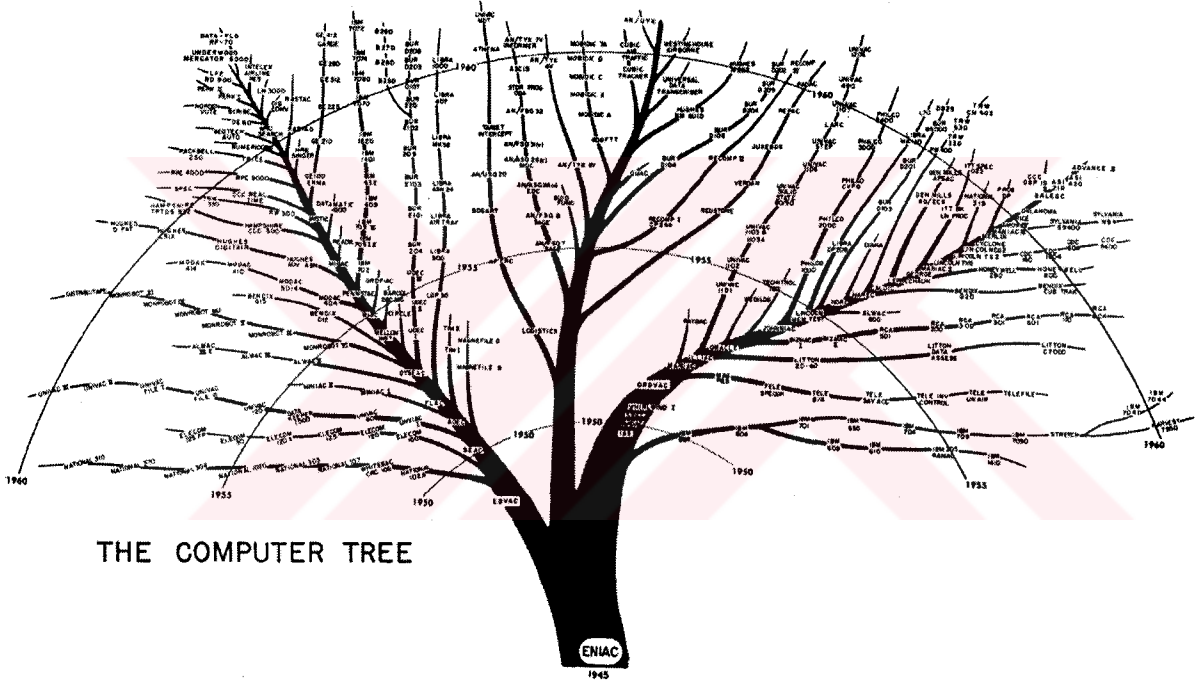
EDVAC, "İkilik Sistem'in" (Binary Coded Decimal) ilk olarak kullanıldığı bilgisayar olma özelliğine sahiptir. Sahip olduğu manyetik bant sürücü, dakikada 75 inç okuyabilmektedir. EDVAC, depolama teknolojisinin önce hız kazandığını, sonra yavaşladığını, sonra tekrar hızlandığını göstermesi açısından iyi bir örnektir (48).

Neumann, çalışmalarına devam ederken, ünlü İngiliz matematikçisi Alan Turing, "Turing Makinesi" ve "Turing Testi" (bir makinenin zeki olup olmadığını belirleyen test) üzerinde çalışmaktadır. Alan Turing, 1959 yılında, bu çalışmalarını "Mekanizmayı ve Zekayı Hesaplamak" (Computing Machinery and Intelligence) adlı makalesi ile kamuoyuna duyurmuştur (Kurzweil, 1998).

Her ne kadar kayıtlı program kavramının yaratıcısı ve bu sayede bilgisayar kavramını bilimsel olarak gerçekleştirilebilir ve evrensel olarak kullanılabilir hale getiren kişi olarak Neumann görülse de, onun Turing'in çalışmalarından etkilendiği söylenmektedir. Bu konuda çalışan bazı bilim adamları, Turing'in çalışmalarının, Neumann'ın çalışmalarından bağımsız olduğunu ileri sürerken, Turing'in çok daha gözü pek ve ileri kavramlar geliştirdiğini belirtmektedirler (Ifrah, 2000).

Hem von Neumann'ın, hem Alan Turing'in isteği (Babbage'dan beri bu konuda çalışan her bilim adamının isteği gibi) evrensel bir makine icat etmek olmuştur. Neumann, daha az kesinleşmiş ve daha karmaşık fikirlere sahip olmakla beraber, Turing'den daha hızlı bir şekilde bunları gerçekleştirebilmiştir. Turing ise fikirlerini uzunca bir süre kendine saklamıştır. Neumann'ı Turing'den bir adım öne çıkaran sebep olarak bunu vermek mümkündür. Ancak kimin kimden etkilendiğini söylemek kesin olarak mümkün değildir, çünkü bu konuda herhangi bir kayıt bulunamamıştır (Ifrah, 2000).

1951 yılında, Eckert ve Mauckley, bu kez, Amerika'da yapılan nüfus sayımının sonuçlarını hesaplamak üzere, "UNIVAC" adı verilen bilgisayarı yapmışlardır. UNIVAC, ilk olarak satışa çıkarılan bilgisayardır ve 46 adet satılmıştır (Şekil 2.6).



Şekil 2.6 Modern bilgisayarların gelişimi (25)

Bell Laboratuvarlarında, 1947 yılında transistörün icat edilmesinden sonra, IBM firması bu teknolojiyi geliştirmiştir. 1956 yılında ilk transistörlü bilgisayarlar IBM firması tarafından piyasaya sürülmüştür. Bilgisayarlar kullanılırken, bunların sadece hesap yapmada değil, bilgi işlemede de başarılı oldukları anlaşılmıştır. Bunun üzerine IBM, mühendisler ve bilim adamları için hesap yapma özelliği daha üstün olan; bankalar ve bürolar için ise bilgi işleme özelliği daha üstün olan iki ayrı model geliştirerek piyasaya sürmüştür (Kurzweil, 1998).

Bu sıralarda, programlama dilleri konsepti geliştirilmeye başlanmıştır. İlk bilimsel programlama dili olan “FORTRAN”, IBM’de geliştirilmiştir (20).

1958’de “Birleştirilmiş Devre’nin” (Integrated Circuit) bulunmasıyla beraber, modern bilgisayarlar bir adım daha ilerlemişlerdir. Bu buluş, bilgisayarın küçülmesinde ve yaygın olarak kullanılmaya başlamasında önemli bir rol oynamıştır (53).

1970’de IBM, bilgisayarlarına “Disket Sürücü” (Floppy Disk) eklemeye başlamıştır. Böylece bilgi depolama yeri artmış ve bilgiye daha hızlı erişim sağlanmıştır (9).

2.2.1 İlk İşletim Sistemleri

İşletim sistemi (operating system), donanımın doğrudan doğruya kullanımından ve yönetiminden sorumlu olan sistem yazılımıdır (53).

İlk işletim istemlerinden olan Unix ile ilgili çalışmalar, 1969 yılında Bell Laboratuvarlarında, Ken Thompson ve Dennis Ritchie’nin başını çektiği bir grup bilim adamları tarafından başlatılmıştır ve günümüzde de devam etmektedir (49).

Diğer bir işletim sistemi olan Multics (Multiplexed Information and Computing Service) zaman paylaşımli bir işletim sistemidir. Multics ile ilgili ilk çalışmalar 1965 yılında başlamıştır ve günümüzde de devam etmektedir. Multics ile ilgili ilk çalışmayı MIT, Bell Laboratuvarları ve General Elektrik beraber yürütmüşlerdir (33).

Multics üzerinde çalışan bilim adamlarından Corbato ve Vssotsky’nin, 1965 tarihli “Multics Sistemine Giriş ve Genel Bakış” (Introduction and Overview of the Multics System) adlı makalelerinde, Multics’in amaçları şöyle sıralanmaktadır:

- Uzaktan uygun terminal kullanımı,
- Elektrik ve telefon sistemlerine benzer devamlı operasyonlar,
- Dosya sistemlerinde yüksek güvenilirlik,
- Bilgi paylaşımını destekleme,
- Sistem düzenlemesi,
- Geniş uygulama alanını destekleme,
- Çoklu program ortamı ve arayüzü destekleme,
- Sistemi teknolojik değişiklikler ve kullanıcı isteklerine göre geliştirebilme.

2.3 Kişisel Bilgisayar Kavramı

Kişisel bilgisayarlar (Personal Computers: PC), insanların her türlü aktivitelerini etkileyen bir devrim olmuştur. Bununla birlikte, kişisel bilgisayarların gelişimi bir devrimden daha çok bir evrim olarak düşünülmelidir (36).

Kişisel bilgisayar, üç önemli elemanın bir araya getirilmesinden oluşmaktadır:

- Fikir
- Donanım
- Yazılım

Kişisel bilgisayarların gelişimi, Xerox Alto araştırmaları ile başlamıştır. Ama bu çalışmalar ticari başarıyı yakalayamamıştır. Ticari başarı elde eden ilk kişisel bilgisayar, IBM firmasının geliştirdiği IBM PC olmuştur. IBM PC bilgisayarlarının genel özelliği, birçok donanım ile çalışabilmesidir. Bu yüzden daha düşük fiyatlar ile satılmış ve yaygınlaşmıştır. Yaygınlaşma 90'lı yıllarda daha da artmıştır. Bunun sebebi, 90'lı yılların sonunda Microsoft firmasının çıkardığı Windows işletim sistemlerinin sadece IBM PC'lerde çalışabilmesidir (20).

2.3.1 Xerox Alto Araştırmaları

“Xerox Palo Alto Araştırma Merkezi” (Xerox PARC), 1969 yılında, bugün “Silikon Vadisi” olarak bilinen yerde kurulmuştur. Bu araştırma merkezinde çalışan bilim adamları 1973 yılında, “Alto” adı verilen ilk kişisel bilgisayarı geliştirmişlerdir. Xerox Alto ilk gerçek bilgisayardır. Bir faresi, grafik arayüzü, Ethernet bağlantısı, ikonları ve pencereleri bulunmaktadır. Alto ile bilgisayarlar küçük ve kullanışlı hale gelmeye başlamışlardır (Kurzweil, 1998).

1981 yılında,PARC'da “Xerox Star” adı verilen ve Xerox Alto'nun ticari versiyonu olan yeni bir bilgisayar geliştirilmiştir. Star, bu zamana kadar geliştirilen bilgisayarların içinde en ileri durumda olan bilgisayardır. IBM'in birkaç ay sonra IBM/PC'yi tanıtmasıyla birlikte, Star'ın başarısı duraklamaya başlamıştır. Yaklaşık 10.000 \$ olan fiyatı, başarısını sınırlayan bir etken olmuştur (15).

Star'ın en önemli özelliklerinden biri, sahip olduğu arayüzdür. Bu arayüz, WYSIWYG (What You See Is What You Get) mantığına sahip bir arayüzdür ve kullanıcının kavramsal modellerini bir biçime dönüştürüp sergileyebilmektedir (17).

Bu arada Xerox PARC'ı ziyaret eden, Apple Bilgisayar'ın kurucularından Steve Jobs, burada geliştirilen fareyi gördükten sonra, Apple tarafından geliştirilmekte olan, "Lisa" ve "Macintosh'a" eklemeye karar vermiştir. Jobs, Apple'ın bu yeni teknolojiyi pazarlamada, Xerox'dan daha başarılı olacağını düşünmektedir. Jobs, bu fikrine Xerox PARC'da çalışan pek çok bilim adamını inandırmayı başarmıştır. Bu bilim adamları, bilgi ve teknolojiyi PARC'dan Apple'a taşıdıktan sonra, Apple 1983 yılında Lisa'yı piyasaya çıkarmıştır. Lisa, Jobs'un PARC'da gördüğü teknoloji üzerine kurulmuştur. Lisa, Star ile aynı nedenden yani pahalı olmasından dolayı başarılı olamamıştır (15) (38).

1984 yılında Apple, "Macintosh'u" piyasaya sürmüştür. Macintosh, Lisa'nın (dolayısıyla Star'ın) tüm özelliklerine sahip ve uygun fiyatlı bir bilgisayar olduğundan, büyük başarı kazanmıştır (36).

Xerox PARC Araştırma Merkezi, yeni buluşlar ve yeni teknolojilerin geliştirildiği önemli merkezlerden biri olmuştur. Fakat, bu buluş ve teknolojilerin pazarlanmasında rakipleri kadar başarılı olamamıştır (15).

2.3.2 Microsoft Windows İşletim Sistemi

"Windows İşletim Sisteminin" ilk fikirleri, 1981'de Xerox Star ile ortaya çıkmıştır. Daha sonra 1984 yılında, Apple Macintosh üzerinde bu fikirler kullanılarak "MacOS" işletim sistemi geliştirilmiştir. Windows'un ilk versiyonu, 1985 yılında Microsoft tarafından piyasaya sürülmüştür (52).

Windows'un ilk versiyonu sağlıklı olarak çalışmadığı için, yeterince başarılı olamamıştır. 1987 yılında geliştirilen "Windows 2", ilk versiyona göre daha başarılı olmuştur. Windows işletim sistemi ilk gerçek başarısını, 1990 yılında piyasaya sürülen Windows 3 ile yakalamıştır. Windows 3 sayesinde IBM/PC, Apple Macintosh'un ciddi rakibi haline gelmiştir. Bu başarının hemen ardından "Windows 3.1" ve "Windows 3.11" geliştirilmiştir, fakat bu versiyonların Windows işletim sisteminin ilerlemesine önemli bir katkıları olmamıştır (19).

"Windows 95", 1995 yılında piyasaya sürülmüştür. Windows 3 ve "MS-DOS" ile pek çok kodu paylaşmakla beraber, iki önemli avantaja sahiptir. İlk olarak, Windows 95, tamamıyla bir işletim sistemidir. Artık MS-DOS'a ihtiyaç kalmamıştır. İkincisi, tamamıyla 32bit bilgisayar mimarisi kullanılarak yapılmıştır (52).

Windows 95'den sonra, özellikle sunucular üzerinde kullanılmak üzere geliştirilen Windows NT (New Technology)", 95'den çok daha güvenilir olmasına rağmen, MS-DOS tabanlı yazılımlarla uyum problemi olduğundan yeterince başarılı olamamıştır (19).

1998 yılında piyasaya sürülen "Windows 98", "Windows 95" ile benzer özellikler taşımaktadır. Tek farkı diskler üzerinde daha fazla bilgi saklamayı sağlayan yeni bir metoda sahip olmasıdır (9).

"Windows 2000'i" piyasaya sürerken, Microsoft'un amacı, Windows 95/8 ve Windows NT'yi birleştiren yeni bir versiyon yaratmak olmuştur. Fakat Windows 2000, 9x serisinin çalıştırabildiği bazı programları çalıştıramadığı için yeterince başarılı olamamıştır (19).

Microsoft, 95/98 serisinin son ürünü olarak, "Windows ME'yi" (Millenium Edition), Windows 2000 NT ve Windows XP' yi piyasaya sürmüştür (52).

Sonuç olarak, bilgisayar teknolojisi geçmişten günümüze sürekli olarak gelişmektedir. Belirli bir dönemde yeni olarak kabul edilen bir teknoloji, çok kısa bir süre içinde eskimekte ve yerini daha yeni, başka bir deyişle daha gelişmiş bir teknolojiye bırakmaktadır.

3. ARAYÜZ KAVRAMI

Bilgisayar teknolojilerinin mimarlık alanındaki en büyük sorunu, mimari yazılımların verimsiz olarak kullanılmasıdır. Bu bölümde, mimari yazılımların daha verimli olarak kullanılması açısından, arayüz kavramı ve arayüz kavramının önemi incelenmiştir.

3.1 İnsan – Bilgisayar Etkileşimi (İBE) nedir?

İnsan-Bilgisayar Etkileşimi, “İBE” (Human-Computer Interaction/ HCI), çeşitli şekillerde tanımlanmaktadır. İBE'nin değişik tanımları mevcuttur. İBE, insanlar tarafından kullanılan interaktif bilgisayar sistemlerinin tasarımı, değerlendirilmesi ve uygulanmasıyla ilgili olarak çalışan bir disiplindir (Preece, 1994). Başka bir tanıma göre İBE, insanların belirli görevleri yerine getirmek için bilgisayar sistemlerini nasıl kullandıklarının, araştırılmasıdır. İBE, bilgisayar ve onu kullanan kişi arasındaki ilişkiyi anlamayı ve bunlar arasındaki ilişkiyi daha verimli, yaratıcı ve rahat bir şekilde dönüştürmeyi amaçlar, İBE genelde bir amaca ulaşmak için yapılan çalışmalarda kullanılır, fakat bilgisayar oyunları gibi herhangi bir amaca ulaşmayı hedeflemeyen çalışmalarda da kullanılmaktadır. (17).

İBE tasarımın önemli özelliklerinden biri, mutlaka spesifik bir içeriğe uygulanmasıdır. Herhangi bir ortam veya amaca uygun olarak yapılan bir İBE tasarımının başka bir ortam veya ortamlara uygulanması mümkün değildir (3). İBE tasarımın bir diğer önemli özelliği, tanımlanabilir bir kullanıcı tipine sahip olmasıdır. Bilgisayar kullanıcıları üç kategoride incelenebilir:

- Uzman kullanıcılar, belirli bir sistem üzerinde detaylı bilgiye sahip olanlar,
- Ortalama kullanıcılar, yerine getirmeleri gereken görevi nasıl yapacaklarını ve bunu yaparken hangi sistemi kullanacaklarını iyi bilenler,
- Acemi kullanıcılar, belirli bir sistemi önceden hiç kullanmamış olanlar.
- Herhangi bir kullanıcı, bir bilgisayar sistemi hakkında uzman düzeyinde bilgiye sahipken, diğer bir sistem hakkında bilgiye sahip olmadığı için acemi kullanıcı kategorisine girebilir. Öyleyse her bilgisayar sistemi için böyle bir kategori oluşturmak mümkündür (3).

3.2 Arayüz nedir?

Arayüz, çok genel olarak, her türlü nesnenin iç ve dış dünyanın birleştiği noktada sahip olduğu ortam şeklinde tanımlanabilir. Kullanıcı arayüzü ise, bir sistemde kullanıcının görebildiği ve sistemi algılamasına yardımcı olan bölümdür. Kullanıcı arayüzü, kullanıcılar ile sistem arasında, sistemi kendisinden beklenen davranışları gösterdiği ve işlevinin açıkça anlaşıldığı bir ilişki kurar. Örnek: gaz pedalı (Jones, 1998).

Bir arayüzün başarılı olarak tanımlanması, o arayüzün sahip olduğu nesnenin anlaşılabilir ve kullanılabilir olması demektir. Anlaşılır ve kullanışlı arayüz tasarımı için iyi bir fikir ve kavramsal model, görsellik, doğru atamalar ve geri besleme özelliklerinin bir arada bulunması gerekir (Norman, 1990).

- **Görsellik:** Modern telefon sistemleri görselliğe az önem verilmiş sistemlere iyi bir örnek oluşturur. Bu sistemlerde bir çağrıyı yakalamak için çok sayıda aktarma numarasını ezberlemiş olmak gerekir. İşte bu modern telefon sistemlerinin en büyük dezavantajlarından biridir.
- **Kavramsal Model:** İyi bir kavramsal model hareketlerimizin sonuçlarını tahmin edebilmemize yardımcı olur. Yaptığımız iş hakkında iyi bir kavramsal model oluşturamazsak, o işi bilinçsiz bir şekilde bir başkasının tarif ettiği kadar yapabiliriz ve bir şeyler aksadığında veya olağan dışı bir durumla karşılaştığımızda ne yapacağımızı bilemeyiz. Bir çok günlük eşyanın (kalem, makas vb) kavramsal modeli oldukça basittir. Ancak bize sunulan kavramsal model yetersiz ve yanlış olduğunda zorluklarla karşılaşabiliriz (Norman, 1990).
- **Doğru Atamalar (mapping):** Doğru atamalar konusunu açıklayabilmek için yine modern telefon sistemlerini örnek verebiliriz. Bu sistem tesadüfen ortaya çıkmış bir sistem değildir. Bir grup insan bu sistemin özelliklerini düşünmüş, işlevleri kontrol etmek için yollar tasarlamış ve hepsini bir araya getirip kullanıcılara sunmuşlardır. Öyleyse, bu sistemi karmaşık kılan durum nedir? Bu sistemi karmaşık kılan durum, aynı düğmelerin hem telefon etmede hem de farklı kombinasyonlarla değişik işlemleri gerçekleştirmekte kullanılmasıdır. Oysa bir arabayı düşündüğümüzde, bir telefon sisteminden daha karmaşık bir yapısı olmasına rağmen, kullanıcının yapacağı işlere atanmış kumandalar belirgindir. Direksiyonu sola çevirince araba sola, sağa çevirince araba sağa gider, yani doğal bir atama kullanılmıştır (Jones, 1998).
- **Geri besleme (feedback):** Kendi sesimizi duymazken birisiyle konuştuğumuzu veya kalemimiz kağıt üzerinde iz bırakmazken yazı yazdığımızı düşünelim; her iki durumda da hiçbir geri besleme yoktur. Bilgisayarda dosya kopyalarken veya program yüklerken görmeye alıştığımız işaret ve mesajlar olmazsa, doğru yolda veya işlemin neresinde olduğumuzu anlayamayız ve sistemde olup bitenleri takip edemeyiz.

3.2.1 Günlük Yaşamdan Arayüzler

Herhangi birisi bir uçağın pilot kabinine oturursa, uçağı kumanda edememesi kimseyi şaşırtmaz. Ancak günlük hayatta kullanılan aletler için aynı durum söz konusu değildir. Örneğin, bir musluğu veya elektrik düğmesini kolayca kullanılabilmesi gereklidir.

Bina girişlerinde kullanılan rüzgarlıkların işlevsel bir yönü vardır. Ama kanatların açılan ve sabit kısımları açıkça işaretlenmezse, içeride kapalı kalma olasılığı olacaktır.

En basit arayüzlerden birisi kapının arayüzüdür. İnsanlar, kapıları genelde sağ elleriyle açarlar. Dikkat edilirse, kapı kollarının da bu kabule göre konumlandırıldığını görülür. İtilecek kapıların itilecek kısımlarının, çekilecek veya kayacak kapıların doğru kısımlarının işaretlenmiş olması gerekir. Doğal olarak yorumlanabilecek, farkına varmak için fazladan çaba gerektirmeyecek tasarımlar yapılmasına dikkat edilmelidir (Norman, 1990).

3.2.2 Arayüz Tasarımının Diğer Disiplinlerle İlişkisi

Arayüz tasarımı disiplinler arası bir çalışma alanıdır. Bu yüzden bir arayüz tasarımcısının insan davranışları, insan ergonomisi, insan psikolojisi ve insanın düşünsel kapasitesi gibi konularda bilgi sahibi olması gereklidir. Günümüzde arayüz tasarımı ağırlıklı olarak bilgisayar yazılımları ile ilgili olduğundan ve bu konu üzerinde geliştirildiğinden tasarım sahibi olmaya ihtiyacı vardır. Arayüz tasarımı ile ilgili bazı disiplinleri şu şekilde sıralayabiliriz (17):

- Ergonomi
- Bilgisayar Bilimi
- Yapay zeka
- Dil bilimi (Linguistik)
- Psikoloji
- Sosyoloji
- Yazılım Mühendisliği
- Tasarım

3.2.3 Arayüz Tasarımının Önemi

İnsan beyni, etrafındaki dünyadan anlam çıkarmaya eğilimlidir. En küçük bir ipucu gördüğünde, bunu açıklamaya, anlamaya ve mantık yürütmeye başlar. Günlük yaşantıda karşılaşılan binlerce eleman arasından iyi tasarlanmış olanları anlamak kolaydır. Bu elemanlar nasıl işlediklerine dair görsel ipuçları taşırlar. İyi tasarlanmamış olanların kullanımı güçtür ve kullanıcılarında psikolojik sorunlar yaratabilir. Arayüz tasarımının önemli bu noktada ortaya çıkar. Kullanıcıların çalışmalarını teşvik eder (Laurel, 1990).

3.2.4 Bilgisayar Biliminde Arayüz Tasarımının Önemi

Geçmişte bilgisayar yazılımlarının yetersiz arayüze sahip olmasından dolayı, verimlilikte büyük kayıplar ve bunun yanı sıra bir görevi gerçekleştirmek için hesaplanan zamanda işlem dışı artmalar görülmüştür.

1980 yılında Amerika’da yapılan bir çalışma yeni bilgisayar sistemlerinin sadece %20’sinin başarılı olduğunu göstermiştir. Geri kalan sistemlerin başarısızlığı büyük ölçüde yetersiz arayüz tasarımı ile ilişkilendirilmiş ve sonunda büyük para, zaman ve güç kayıpları getirmiştir (17).

Günümüzde, ihtiyaçları karşılayan, yeterli bir arayüz tasarımı bilgisayar sistemlerinin verimliliğini, kullanılabilirliğini ve dolayısıyla popüleritesini arttırmaya yarayan en önemli özelliklerden birisidir. Bilgisayar sistemlerinin hem yazılım, hem donanım alanında geliştirilmesiyle, yeni bir kullanıcı tipi ortaya çıkmıştır. Bu yeni kullanıcı tipi, kullanım kolaylığı olan yeni arayüzler talep etmektedir. Bu yüzden geleneksel diyebileceğimiz çalışmalar, özellikle yazılım geliştirilmesi alanında, arayüz tasarımı ve geliştirilmesine de içine alarak yeniden yapılanmaya başlamıştır. Arayüz tasarımı çalışmaları bitmiş bir ürüne yapıştırılmış etiket gibi görülmekten kurtulup, ürünün tasarımının her aşamasında etkili olan bir konum kazanmıştır. Arayüz tasarımı gelecek birkaç yıl içerisinde çok daha önemli hale gelecektir, hatta pek çok sistem için yasal ve zorunlu bir gereklilik olacaktır. Avrupa Topluluğu’nun bu konuyla ilgili bazı çalışmaları olduğu bilinmektedir (17).

Gelecekte, arayüz tasarımının önemi şu alanlarda artacaktır;

- Yazılım geliştirme süreci ve sistem tasarımı metotlarının bir parçası olarak,
- Yazılımlar için yasal ve zorunlu bir gereklilik olarak,
- Ürünlerin seçiminde ve değerlendirilmesinde bir kullanım kriteri olarak,
- Kullanıcıların artırılmasında bir pazarlama stratejisi olarak.

3.2.5 İyi Bir Arayüz Tasarımı Nasıl Olmalıdır?

İyi tasarlanmış bir arayüz kullanıcıların ihtiyaçlarına cevap vermeli, kullanıcıların yeteneklerine ve öğrenme kapasitelerine uygun olmalı ve kullanıcıya memnuniyet ve verimlilik sağlamalıdır. İyi tasarlanmış bir arayüzün öğrenilmesi ve kullanılması kolay olmalıdır. Kullanıcılar “acaba nerede ve ne yanlış yapacağım” korkusuna düşmeden arayüzü kullanabilmeli ve daha verimli çalışabilmeleri teşvik edilmelidir (17).

İyi bir arayüz, mümkün olabilecek en verimli şekliyle kullanıcı tarafından algılanabilmeli ve kullanıcının ihtiyaçlarına cevap verebilmelidir. Kullanıcı neyi talep ediyorsa onu karşılamak ve kullanıcının görevini yerine getirebilmesi için gereken sistemi en anlaşılır şekliyle tasarlamak arayüz tasarımının en önemli amacıdır (Shneiderman, 1998).

3.2.6 Arayüz Tasarımının Genel Prensipleri

Arayüz tasarımının genel prensipleri şu şekildedir:

- **Doğallık (Naturalness):** Kullanıcı, kullandığı yöntemin yerine getirmesi yeterli olan göreve göre sürekli olarak değişmesini istememektedir. Ayrıca, kullanıcının kendi tasarladığı düzen ile kullandığı arayüzün düzeni arasında oluşabilecek tutarsızlıklar, arayüzün verimli bir şekilde kullanılması olasılığını azaltır. Bu yüzden, arayüz tasarımında doğal bir akış oluşturulmasına dikkat edilmelidir (17).
- **Tutarlılık (Consistency):** Kullanıcı kullandığı arayüzün tutarlı olmasını, beklentilerinin karşılanacağını teminatı olarak görmelidir. Kullanıcı, arayüzün tutarlı olduğuna inandığı zaman, arayüzün kullanılma olasılığı artar (17).
- **Gereklilik (Non-redundancy):** Arayüzün, kullanıcı neye ihtiyaç duyuyorsa onu karşılaması anlamına gelir. Kullanıcı, minimum bir bilgi ile sistemi kullanabilmeli ve bir görevi yerine getirebilmek için kendini zorlayarak bir şeyler öğrenmek zorunda kalmamalıdır (17).
- **Destekleyicilik (Supportiveness):** Arayüz içinde bir takım yardımcı öğeler bulunmalıdır. Bu yardımcı öğelerin gerçekten aydınlatıcı ve açık olmasına dikkat edilmelidir, aksi takdirde kullanıcının kafası daha çok karışabilir. Bir takım yardım öğeleri ile destekleneceğini bilmek, kullanıcının arayüze duyduğu güveni artırır ve öğrenme sürecini azaltır (17).

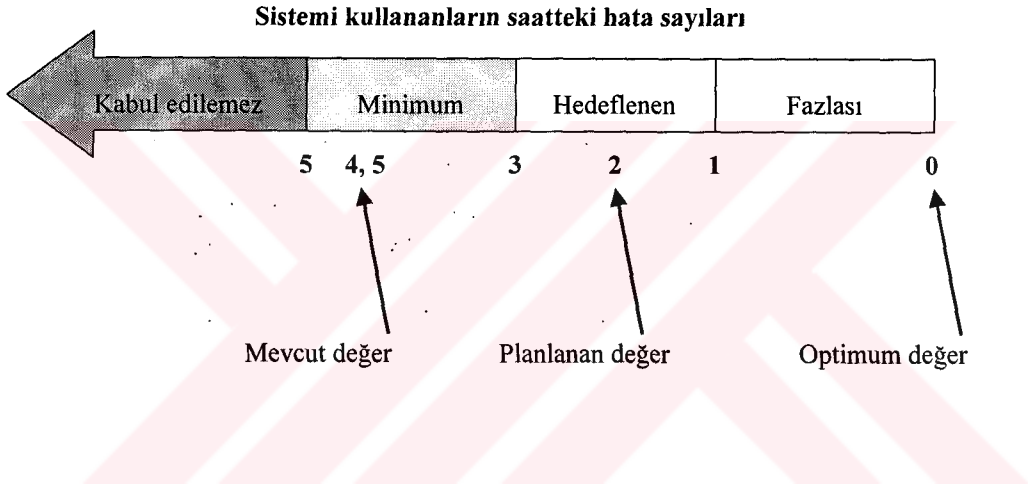
- **Esneklik (Flexibility):** Her kullanıcı bir arayüzü en verimli şekilde kullanabilmeyi istemektedir. Verimli kullanım, kullanıcının bilgi düzeyi ile ilgili olduğu kadar, arayüzün kullanıcının bilgi düzeyine indirgenebilmesi ile de ilgilidir. Bu şekilde tasarlanan bir arayüzü, deneyimli kullanıcılar kadar, yeni kullanıcılar da rahatlıkla kullanabilir. Dolayısıyla, arayüzün kullanılabilirliği ve yaygınlığı artar (17).
- **Açıklık (Clarity):** Pek çok kullanıcı, sıklıkla arayüzlerin yeterli şekilde anlaşılmadığında şikayetçi olmaktadır. Bu yüzden, bir arayüz içinde sıklıkla kullanılan terimlerin açıkça anlaşılır bir şekilde yer almasına dikkat edilmelidir (42).
- **Okunabilirlik (Readability):** Arayüz tasarımında, akıcı, tarafsız ve kolayca anlaşılabilir bir dil kullanılması anlamına gelmektedir. Çünkü klişeleşmiş ve satışa yönelik olarak kullanılan terimlerden oluşan bir dilin, kullanıcı üzerinde bilişsel bir baskıya yol açtığı bilinmektedir (42).
- **Global Uyumluluk (Global Compatibility):** Tasarlanan arayüzün, dünyanın çeşitli bölgelerinde kullanılan sistemlerle uyumlu olması gerekmektedir (42).

3.2.7 Arayüz Tasarımında Tasarım Süreci

Arayüz tasarımı, ürün piyasaya çıkmadan hemen önce yapılmamalıdır. Ürün tasarımı ile birlikte arayüz tasarımı çalışmaları başlamalıdır. Ürünün piyasaya verilmesine kadar geçen süreç şu adımlardan oluşur:

- **Kullanıcıyı tanımak:** Arayüzün hedef kullanıcıların ve ne amaçla kullanılacağını incelenmesi gereklidir. Bunun için tasarımcılar en azından kullanıcıların bulunduğu yere gitmeli ve fikir edinmelidirler. Kullanıcıların “ne isteyebilecekleri” tartışmasıyla zaman kaybetmek yerine, doğrudan onlarla konuşup “ne istediklerini” öğrenmek kuşkusuz daha etkili bir yöntemdir. Kullanıcıların bireysel farklılıkları ve yaptıkları işlerdeki değişiklik dikkat edilmesi gereken iki önemli faktördür. Kullanıcı, yalnız ürünü kullanan kişi değil, o ürünün kullanımından etkilenecek herkes olarak düşünülmelidir (Nielsen, 1994).
- **Rekabet analizi yapmak:** Tasarlanacak ürüne/sisteme benzeyen, piyasada mevcut diğer rakip ürünler aslında iyi bir prototip görevi görebilirler. Bunları sistematik olarak değerlendirmek ve bu ürünlerle kullanıcı deneyleri yapmanın yanı sıra, birden fazla rakip ürün varsa, bunlar arasında bir karşılaştırma yapmak da mümkündür. Bu tür bir çalışma, yeni ürün için bilgi alınmasına yarayabilir. Ürünün piyasası ile ilgili periyodik yayınların takip edilmesi de diğer ürünler hakkında karşılaştırmalı bilgiler sağlayabilir. Bunlar arasında seçilecek ürünler arasında yapılmış tasarımların taklit edilmesi anlamına gelmemektedir.

Önemli olan daha önceden yapılanları inceleyerek daha iyi tasarımlar ortaya koyabilmektir (Nielsen, 1994). Hedeflerin belirlenmesi özellikle daha önceden piyasaya sürülmüş ürünler veya belirli bir rakibi olan ürünler için kolaydır. Daha önce hiç kullanılmamış olan sistemler için hedef belirlemek daha zordur. Bir yaklaşım uzmanlara danışmaktır. Kullanıcılara sormak da bir yöntem olabilir, ancak birincisi kadar güvenli olmayabilir. Kullanıcıların ve yaptıkları işlerin incelemesi sonucu, arayüzün hangi hedefe öncelik tanıyacağına karar verilebilir (Nielsen, 1994) (Şekil 3.1).



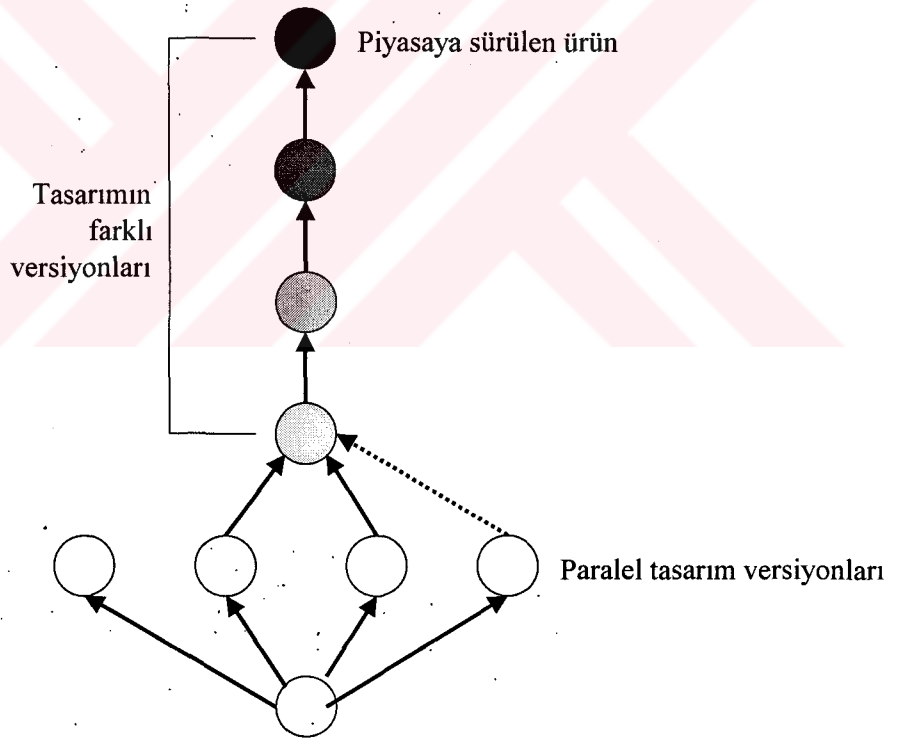
Şekil 3.1 Rideout (1991) tarafından geliştirilen kullanılabilirlik hedef çizgisi

- **Paralel tasarım:** Tasarım sürecine, birden fazla tasarımcının ön tasarım yaptığı paralel tasarım aşaması ile başlamak iyi bir fikir olabilir. Paralel tasarımın amacı fazla detaya girmeden, üzerinde çalışılabilir alternatif çözümler elde etmektir. Paralel tasarım aşamasına üç veya daha fazla tasarımcı katılabilir. Bazı önemli projelerde, paralel tasarımın, birkaç ekip tarafından, hangi alternatifin piyasaya sürüleceğine karar verilmesine kadar sürdürüldüğü bilinmektedir (17).

Paralel tasarımda tasarımcıların birbirlerinden bağımsız olarak çalışmaları gerekir, çünkü mümkün olduğu kadar farklı fikirlerin ortaya çıkması istenmektedir. Bazen aynı temel fikir üzerine kurulmuş tasarımlar bile tamamen farklı detaylandırıldıklarından farklı tasarımlara dönüşebilmektedir.

Paralel tasarımın bir başka yöntemi de yaygın paralel tasarım olarak bilinmektedir. Bu yöntemde birden fazla tasarımcıdan aynı tasarımın farklı yönleri üzerinde çalışmaları istenir. Böylece farklı yaklaşımları deneme fırsatı doğar. Daha sonra, istenirse bu çalışmalar tek bir çözüm içersine entegre edilebilir.

Özellikle ilk defa tasarlanacak olan arayüzler için paralel tasarım iyi bir araçtır. Daha önce uygulanmış veya piyasada rakiplerinin bulunduğu arayüzler için ise keşfedilmemiş fikirler ortaya çıkartmakta kullanılabilir. Paralel tasarımda, birden fazla tasarımcı mümkün olabilecek çözümleri daha kısa sürede inceleyebileceği için, ürün daha kısa sürede geliştirilebilir ve kullanıcılara sunulabilir hale gelir (17) (Şekil 3.2).



Şekil 3.2 Orijinal tasarım konsepti (17)

- **Katılımcı tasarım:** Her ne kadar tasarıma başlamadan önce ‘kullanıcıyı tanıyın’ tavsiyesine uyulsa da tasarım süreci başladıktan sonra kullanıcılarla ilgili bilinmeyenlerle karşılaşılacaktır. Tahminde bulunmak yerine tasarımcılar hedef kitleyi temsil eden kişilere ulaşabilmelidirler.

Kullanıcılar düzenli toplantılarla tasarım sürecine dahil edilebilirler. Bu tür katılımcılara “Konu Uzmanları” (Subject Matter Experts) de denmektedir. Doğal olarak, kullanıcılar tasarımcı olmadıkları için onlara sıfırdan tasarım yaptıramazsınız. Yapılan çalışmaları kullanıcıların anlayabileceği bir formatta onlara gösterip fikirlerini alabilirsiniz. Henüz çalışabilir bir prototip olmadığı için, bir maket hazırlanabilir (17).

Kullanıcıların bu aşamada söylediklerini de dikkatli bir şekilde kullanmak gerekir. Bir araştırmada, programın bir özelliği hakkında kullanıcıların anlatılanlara göre yaptıkları değerlendirmeler ile, aynı özelliği kullandıktan sonra yaptıkları değerlendirmenin hiç bir benzerliği olmadığı gözlemlenmiştir (Schneiderman, 1998).

Katılımcı grubu bir kaç kez yenilemek, konuya fazlasıyla aşina olup, tasarımcılar gibi düşünmeye başlayan katılımcılarla çalışmak riskini azaltacaktır. Ne kadar farklı kullanıcı proje hakkında fikrini söylese o kadar geniş bir kullanıcı yelpazesine ulaşılmış olacaktır.

- **Tüm arayüzün koordinasyonu:** Tutarlılık, arayüz tasarımı için önemli bir kavramdır. Bir tasarımın her türlü medyada yer alabilecek anlatımları birbirleri ile tutarlı olmalıdır. Çoğu zaman farklı gruplar veya departmanlar tarafından hazırlanan ürün bileşenleri arasında uyumsuzluklar gözlenmiştir.

Aynı ürünün uzun bir zaman dilimi içerisinde kullanıma sunulabilecek farklı versiyonları bile birbirleriyle tutarlı olmalıdır. Birden fazla değişik ürün geliştiren şirketlerin, belirli bir arayüz standartları olduğu bilinmektedir (Schneiderman, 1998).

Tüm arayüzün tutarlılığını sağlamak için merkezi bir otorite oluşturulabilir. Bu tek bir kişi olabileceği gibi, büyük projelerde bir komite de olabilir. Bu kişi veya kişiler arayüz tasarımında hem genel standartların hem de o tasarıma özgü standartların uygulanmasından sorumlu tutulacaktır.

- **Tasarım prensiplerinin uygulanması:** Tasarım prensipleri, bir programın geliştirilme sürecinde kullanılacak olan yol gösterici kuralların toplamıdır. Bir proje içinde bir kaç değişik düzeyde tasarım prensipleri kullanılabilir. “Genel Tasarım Prensipleri” tüm arayüzler için geçerli olan doğallık, tutarlılık, gereklilik, destekleyicilik, esneklik prensipleridir. “Kategoriye Özel Prensipler” sistemin türüne bağlı olan prensiplerdir. Bir de “Ürüne Özgü Prensipler” olabilir ki bunlar tasarlanmakta olan projeye özgü prensiplerdir.

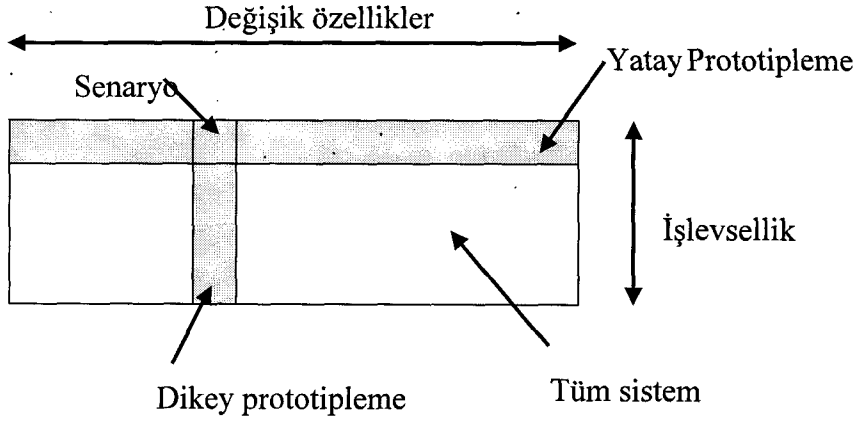
Standartlarla prensipler arasındaki temel fark, standartların arayüzün kullanıcıya nasıl görünmesi gerektiği, prensiplerin ise arayüzün nasıl kullanılabilir hale getirileceği üzerinde durmalarıdır. Örneğin kullanıcılara istenmeyen bir durumdan geri dönebilme imkânı sağlamak bir tasarım prensibidir. Bunun için özel bir düğme, tuş, ikon vb. bir şeyin gerektiği ise bir standarttır (Laurel, 1990).

- **Prototip çalışmaları:** Ön tasarım çalışmalarında arayüzü uygulamaya geçmek iyi bir yöntem değildir. Gerçek ürün veya sistemden önce, hem daha çabuk hem de daha ucuza kullanıcılar tarafından değerlendirilebilecek prototipler geliştirilebilir.

Az sayıda ama işleyen özelliklerle, veya aslında işlemeyen ama tüm özellikler kullanılarak bir prototip hazırlanabilir. Özelliklerin sayısının azaltılmasına “Dikey Prototipleme” denir. Böyle bir prototip tüm sistemin çalışan ancak küçük bir kısmının tam anlamıyla denenmesinde kullanılabilir.

İşlerliğin azaltılmasına ise “Yatay Prototipleme” denir. Arayüzün tamamı simüle edilir ancak kullanıcılar bir işlem yapamazlar. Yatay prototipler arayüzün tamamı hakkında bir fikir edinebilmek için hazırlanan ara ürünlerdir (Laurel, 1990).

Bir de hem özellik sayısını hem de işlerliği minimumda tutup önceden belirlenmiş bir senaryoya göre kullanıcının sistemi denemesi istenebilir (Şekil 3.3).

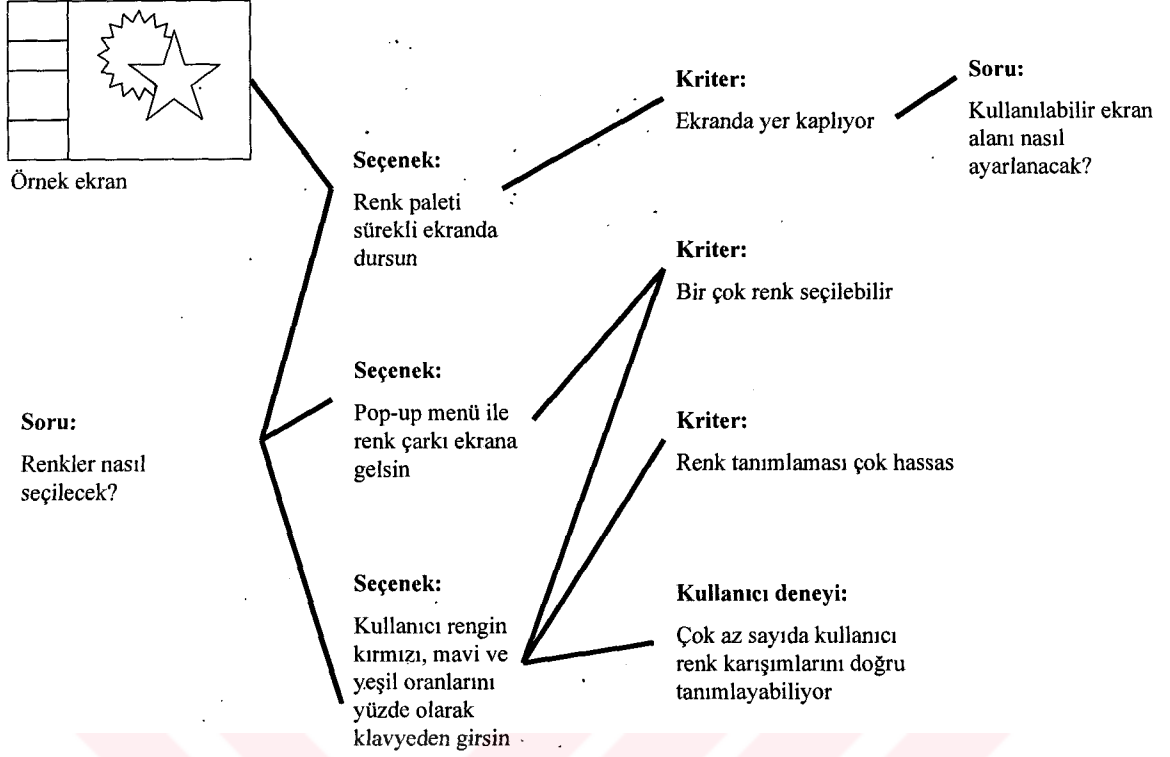


Yatay prototip çalışması tüm özellikleri korurken işlevselliği azaltabilir, dikey prototip çalışması ise işlevselliği korurken sistemin özellik sayısını azaltabilir

Şekil 3.3 Prototip çalışmasının iki boyutu (Laurel, 1990)

- **Değerlendirme:** Arayüz değerlendirmesinin çeşitli yöntemleri vardır. Kullanıcı deneyleri ve uzman kişilerin görüşlerinin toplanmasıyla yapılabilir.
- **Yeniden tasarım:** Kullanıcılarla yapılan deneylerin sonuçlarına bağlı olarak arayüzün farklı versiyonları tasarlanabilir. Deney kayıtları kullanıcıların duraksadığı veya sıkça hata yaptığı yerleri gösterecektir. Bu noktada alternatif çözümler üretebilmek için, yaratıcılık ve biraz da şansa ihtiyaç vardır (Schneiderman, 1998).

Bazen yapılan değişiklikler, o sorunla hiç karşılaşmamış olan kullanıcıları rahatsız edebilir. Yani yeniden tasarımla daha iyi sonuçlar elde edileceğine daha kötü sonuçlar elde edilir. Bu nedenle yapılan değişikliklerin ve neden bu kararların alındığının, ilk çıkış noktasının ne olduğunun unutulmaması için yazılı olarak saklanması gereklidir (Şekil 3.4).



Şekil 3.4 Renk seçimi ile ilgili örnek ekran (Schneiderman, 1998)

- **Kullanıcılardan geri besleme:** Bir ürün kullanılmaya başladıktan sonra, bir sonraki versiyonda veya yeni bir üründe yararlanılmak üzere, kullanıcılardan bilgi toplanmalıdır. Nasıl mevcut sistemlerin incelenmesi yeni ürün için bir bilgi kaynağı ise, yeni piyasaya sürülmüş bir ürünün incelenmesi de öyledir (Schneiderman, 1998).

Yapılan pazar araştırmalarından veri elde edilebilir. Kullanıcılarla yerinde anket yapılabilir. Kullanıcı şikayetleri incelenebilir. Önceden yapılan finanssal etki tahminleri, gerçek ölçümlerin sonuçları ile karşılaştırılabilir.

3.2.8 Kullanılabilirlik (Usability) nedir?

Kullanılabilirlik, bir sistemin etkili (effectiveness), güvenli ve bir şekilde öğrenilebilmesi veya kullanılabilmesi olarak tanımlanmaktadır (Preece, 1994). Bir başka tanıma göre kullanılabilirlik, kullanıcıların ürünü/sistemi, görevlerini yerine getirebilmek için, çabuk ve kolay bir şekilde kullanabilmesidir (Dumas, Redish, 1993).

Kullanılabilirlik, bir sistemin kolay öğrenilmesi ve kolay kullanılması açısından düşünülmektedir. Buna rağmen, belirli bir sistemin kullanılabilir olması, o sistemin yeterliliğinin ölçütü değildir. Aynı zamanda, kim tarafından kullanılacağını da tanımlanması gereklidir (Lindgaard, 1994).

3.2.8.1 Kullanılabilirlik Tasarımı

Günümüzde “kullanılabilirlik” (usability) terimi, “kullanıcı dostu” (user-friendly) teriminin yerine tercih edilmektedir. Kullanılabilirlik aslında bir sistemin “kabul edilebilirliği” (system acceptability) gibi daha kapsamlı bir konunun alt başlığı olarak düşünülmektedir. Kabul edilebilirlik, yalnız o sistemin kullanıcılarını değil, kullanıcıya bağlı olan herkesi ve her şeyi ilgilendirir. Bir sistemin kullanılabilirliği, o sistemin yararlılığı (usefulness) ve işlevi (utility) ile de ilgilidir. Yararlılık, ürünün istenilen görevi yerine getirmekte kullanılıp kullanılmayacağı; işlev ise ürünün istenileni yapıp yapamayacağıdır (Lindgaard, 1994) (Nielsen, 1994).

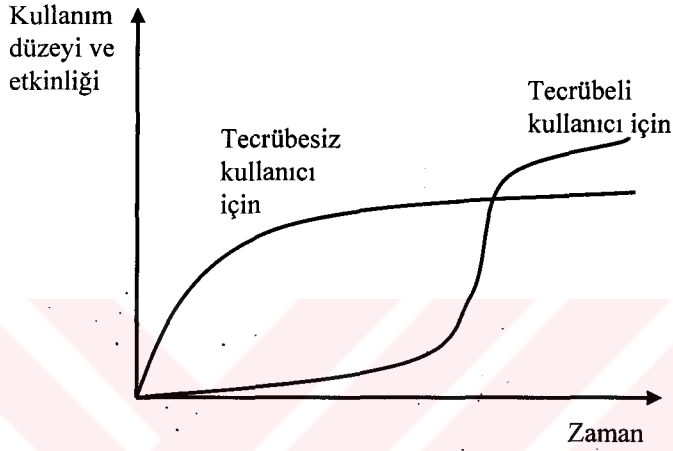
“Kullanıcı dostu” (user-friendly) teriminin yerine daha uygun yeni terimler bulunmasıyla ilgili çalışmaların sonucunda, son yıllarda, aşağıdaki terimlerin kullanımı da yaygınlaşmıştır (Nielsen, 1994).

CHI	Computer-Human Interaction / Bilgisayar-İnsan Etkileşimi
HCI	Human-Computer Interaction / İnsan-Bilgisayar Etkileşimi
UCD	User-Centred Design / Kullanıcı Merkezli Tasarım
MMI	Man-Machine Interface / İnsan-Makine Arayüzü
HMI	Human-Machine Interface / İnsan-Makine Arayüzü
OMI	Operator-Machine Interface / Operatör-Makine Arayüzü
UID	User-Interface Design / Kullanıcı Arayüzü Tasarımı
HF	Human Factors / İnsan Faktörleri

3.2.8.2 Kullanılabilirlik Tasarımının İlkeleri

Kullanılabilirlik tek boyutlu değil, çok boyutlu bir terimdir, birçok birleşenden oluşur. Şu özelliklerin bir toplamı olarak düşünülmelidir:

- **Öğrenilebilirlik:** Öğrenilebilirlik, temel kullanılabilirlik faktörüdür, çünkü bir kullanıcının sistemle yapacağı ilk iş onu kullanmayı öğrenmek olacaktır. Öğrenme kolaylığı sistemi bilmeyen bir kullanıcının yukarıdaki öğrenme eğrisinin ilk kısmındaki deneyimi ile ilgilidir (Şekil 3.5). Bu kısımda dik bir tırmanış gösteren sistemleri veya ürünleri kullanmak kolay olacaktır ve kullanıcılar mantıklı bir süre içinde sistemi veya ürünü etkin bir şekilde kullanabilir hale gelecektir.



Şekil 3.5 Öğrenme eğrisi (Schneiderman, 1998)

Öğrenmenin kolaylık derecesini ölçmek, ürünün ilk defa kullanımında, çok zor olmamaktadır. İlk defa kullanımda, bu sistemi kullanacak olanların, daha önceden belirlenmiş bir düzeye ne kadar zamanda ulaştıklarını ölçmek mümkündür. Burada önemli olan ürünü önceden kullanan ve daha önce hiç kullanmayan kişiler arasında da ayırım yapmaktır.

Öğrenebilirlik, kullanıcıların verilen işi ne kadar sürede yapabilir duruma geldiklerinin veya belirli bir sürede belirli sayıda işi yapıp yapamadıklarının gözlenmesiyle ölçülebilir. Diğer taraftan, kullanıcıların, arayüzü tüm bileşenlerini öğrenmeden de sistemi veya ürünü kullanabileceklerini unutmamak gerekir. Çoğu zaman, kullanıcılar sistemi kendilerine gereken kısmı öğrenir öğrenmez kullanmaya başlarlar. Bu yüzden, sistemin tümünü öğrenmeleri için değil, belirli bir işi yapabilmeleri için geçen süreye önem verilmelidir (Schneiderman, 1998).

- **Etkin Kullanım:** Etkinlik, kullanıcının stabil bir öğrenme düzeyine ulaştığında gösterdiği performans olarak tanımlanabilir. Ürünün niteliğine bağlı olarak stabil düzeye ulaşma süresi değişebilir. Bir ürün için aylar alırken, bir diğeri için sadece birkaç gün alabilir.

Deneyimli kullanıcıların etkinliğini ölçmek için önce deneyimli kullanıcılar bulmak gerekir. 'Deneyimli' tanımı çoğunlukla informal olarak yapılır ve kullanıcılar kendileri deneyimli olduklarını söyledikleri için veya belirli bir süreden daha uzun zamandır o ürünü kullandıkları için deneyimli sayılırlar.

Test gruplarından da deneyimli kullanıcılar elde edilebilir. Öğrenme eğrisinin en başından itibaren kullanıcılar takip edilir ve belirli bir işi hep aynı sürede yapmaya başladıkları aşama, yani deneyimli hale geldikleri aşama belirlenir (Preece, 1994).

Yani, etkinliğin tipik ölçü yöntemi belirli bir deneyim tarifi belirlemek, bu deneyimdeki kullanıcılardan oluşan bir grup toplamak ve bu insanların bazı tipik görevleri ne kadar sürede yaptıklarını takip etmek olacaktır.

- **Hatırlanabilirlik:** Aslında öğrenilebilirlik yolunda kat edilen gelişmeler bir arayüzün hatırlanmasını da kolaylaştıracaktır. Ama bir ürünü ilk defa kullanmak ile uzun bir aradan sonra tekrar kullanmak arasında kullanılabilirlik açısından bir fark vardır.

Hatırlanabilirliği ölçmek için iki yöntem vardır. Birincisi belirli bir süre ürünü kullanmamış kişilere tekrar ürünü kullandırtarak ne yaptıklarını gözlemlemektir. İkincisi ise, kullanıcılara, bir test oturumundan sonra, programdaki komutlarla ilgili sorular sormak ve doğru yanıtların oranını belirlemektir. Birinci yöntem ikincisinden daha etkili bir yöntemdir. Zira modern arayüzler görselliğe dayandığı için, kullanıcıların ürünle ilgili tüm detayları hatırlamasalar bile, ürünü rahatça kullandıkları gözlenmiştir (3).

- **Sübjektif Tatmin:** Sübjektif tatmin ürünün kullanımının keyifli olup olmadığı ile ilgilidir. Bu özellikle eğlence amaçlı olarak tasarlanan ürünler için önemlidir. Çünkü, kullanıcılar böyle bir ürünle sıkılmadan, eğlenerek uzunca zaman geçirmek isteyebilirler.

Prensipte bir arayüzün hoşta gidip gitmediğini objektif olarak ölçmek de mümkündür. Nabız, göz bebeklerinin büyüklüğü, tansiyon ve kandaki adrenalin oranı gibi fizyolojik testler yapılabilmektedir. Fakat bu tür deney ortamları rahatsız edici olduğu için tercih edilmemektedir (17).

Bu yüzden en iyisi doğrudan kullanıcıya sormaktır. Bir test oturumundan sonra kullanıcılar arasında bir anket yapılarak buradan toplanan sonuçlar değerlendirilebilir. Gerçek hayattaki kullanıcılara da benzer anket soruları sorulabilir. Ancak, özellikle yeni ürünler için, kullanıcı bunu denemeden önce soru sormak sakıncalıdır. Kullanıcı beğenmediği en ufak bir şeyden ötürü ürünü kullanmayı reddedebilir. Bu yüzden ürünün çekiciliğinin de ölçülmesi, özellikle pazarlama açısından yararlı olabilir.

Sübjektif tatmin anketleri genellikle çok kısadır. Kullanıcılardan sistemi 1'den 5'e veya 1'den 7'e kadar bir değer vererek, "Likert Skalası" veya "Anlamsal Değişkenlik Skalası" (Semantic Differential Scale) üzerinden değerlendirmeleri istenir (17).

Likert Skalasında ankette 'bu ürünü kullanmak hoşuma gitti' benzeri cümleler bulunur ve kullanıcılara bu değerlendirmelere ne ölçüde katıldıkları sorulur. 1'de 5'e kadar olan bir skalada 1=kesinlikle katılmıyorum, 2=kısmen katılmıyorum, 3=çekimser, 4=kısmen katılıyorum, 5=kesinlikle katılıyorum anlamlarına gelir (17).

Anlamsal Değişkenlik Skalasında ise, kullanıcıdan, ürünü birbirinin zıttı iki değerlendirme arasında bir yere yerleştirmesi istenir (Şekil 3.2)

Lütfen sistem hakkında aşağıda yazılmış olanlara ne derece katıldığınızı belirtin:

- "Sistemin nasıl kullanılacağını öğrenmek çok kolaydı"
- "Bu sistemi kullanmak çok sinir bozucu bir deneyimdi"
- "Bu sistemin bana çok yüksek düzeyde verimlilik sağladığını hissediyorum"
- "Sistemde yaptıklarımın bir çoğunun yanlış olmasından endişe duyuyorum"
- "Bu sistem ihtiyacım olduğunu düşündüğüm her şeyi yapabilir"
- "Bu sistemle çalışmak çok güzel"

Likert skalası kullanılarak yapılmış bir anket. Puanlama 1 ile 5 arasında yapılacaktır.

Bu sistem hakkındaki izlenimlerinizi en iyi yansıtan yeri işaretleyin:

- | | |
|-----------------------|-----------------|
| Güzel | Sinir bozucu |
| Tamamlanmış | Eksikler var |
| Uyumlu | Uyumsuz |
| Basit | Karmaşık |
| Kullanışı hızlı | Kullanışı yavaş |
| Güvenilir | Güvenilmez |

Anlamsal değişkenlik skalası kullanılmış bir anket

Şekil 3.2 Likert ve anlamsal değişkenlik skalaları (17)

3.2.9 Arayüz Tasarımında Değerlendirme

3.2.9.1 Değerlendirme (Evaluation) Nedir ?

Değerlendirme (usability evolution, usability testing, user testing) ürünün, kullanıcıların ihtiyaçlarına cevap verip vermediği saptama sürecidir. Genellikle UCD (User-Centred Design/ Kullanıcı Merkezli Tasarım) tekniği kullanılmaktadır. Değerlendirme testleri, ürünün beklentilere cevap verip vermediğini ve problemlerin neler olduğunu belirleme için kullanılmaktadır. Değerlendirme testleri ile tüm problemleri belirlemek mümkün olmamaktadır. Bununla beraber, tasarımın uygunluğunun incelenmesi, kullanıcıların beklentilerinin belirlenmesi gibi konularda yeterli sonuçlara ulaşılabilmektedir (17).

Değerlendirme testleri resmi ve resmi olmayan şekillerde yapılmaktadır. Resmi olarak yapılan testlerde daha katı bir uygulama yöntemi kullanılır. Bu şekilde kullanıcı davranışları ile uygunluk gibi konuların analizi daha başarılı olmaktadır. Resmi olmayan şekillerde yapılan değerlendirme testlerinde ise istatistiksel resmi sonuçlar elde edilmemekle birlikte, tasarımların kullanılabilirliği ve özellikleri ile ilgili bilgiler elde edilebilir (17) (Jones, 1998).

Arayüz tasarımında değerlendirme yapmanın avantajları şunlardır:

- Potansiyel kullanıcıların reaksiyonlarını ölçebilme,
- Problemler ve problemler kısımları ayırt edebilme,
- Sistemin verimli çalışıp çalışmadığını anlayabilme,
- Tasarımla ilgili kritik noktaları değerlendirebilme,
- Kullanıcı önerileri aracılığıyla yeni fikirler edinebilme,
- Kullanıcı önerilerini karşılaştırabilme,
- Kullanıcıların sisteme olan güvenlerini arttırabilme,
- İleriki çalışmalar için kullanıcıdan destek alabilme,

3.2.9.2 Değerlendirme Yöntemleri

Değerlendirme yöntemleri çok çeşitli olup, hiçbirisi diğerinden tam anlamıyla bağımsız değildir. Değerlendirme yöntemleri arasındaki başlıca farklılık, bilgi toplama yollarının değişmesidir. Bu yöntemlerin başlıcaları şunlardır:

- **Kullanım Şeklini Gözleme:** Bu yöntem, kullanıcının ürünle veya prototip ile ilişki kurduğu anları izleme şeklinde gelişen bir yöntemdir. Kullanıcının kendi çalışma alanında veya bir laboratuarda gerçekleştirilir. Bilgi toplama ve bilgi analizi, gözlemcinin kullanıcıyı izleyerek notlar alması ve/veya ses, görüntü kayıtları ile gerçekleştirilir.

Bu yöntemde, kullanıcılara normal çalışma saatleri içinde yapabilecekleri bir görev verilir. Gözlemci bu süre zarfında gördüğü her şeyi not eder ve kayda alır. Sonuçta görevi yerine getirmenin ne kadar süre aldığı, ne şekilde gerçekleştirildiği, ortalama iş süresinin ne olduğu vb. sorulara yanıtlar alınabilir.

Gözleme yöntemi basit ve akılcı bir yöntem olmakla beraber, görüldüğü kadar doğru sonuçlar vermeyebilir. Gözlemci gördüklerini yanlış yorumlayabileceği gibi, kullanıcı da gözlemlendiğini bildiğinden değişik şekillerde davranabilir (Jones, 1998).

- **Yüksek Sesle Düşünme:** Sözlü anlatım olarak da tanımlanan bu yöntem, gözleme yöntemi sırasında ekstra bilgi edinmek için kullanılan bir yöntemdir. Gözlemci, kullanıcıları düşüncelerini yüksek sesle dile getirmeleri için cesaretlendirir. Bu, gözlemciye kullanıcının nasıl plan yaptığı, işler ters gittiğinde nasıl tepki verdiği, ürünü ne derece anladığı gibi konularda ipucu verir (Schneiderman, 1998)
- **Görüşme ile Kullanıcı Görüşü Toplama:** Kullanıcının performansını inceleme tek başına fazla sonuç vermeyebilir, Bu yüzden kullanıcılar ile kullandıkları ürün hakkında görüşmeler yapılır. Burada kullanıcılara kullandıkları üründen hoşlanıp hoşlanmadıkları, teknolojisini beğenip beğenmedikleri gibi konularda sorular sorulabilir.

Bu yöntem kullanılacaksa, sorulacak soruların dikkatli bir şekilde düşünülerek hazırlanması gerekir. Görüşme yöntemi, esnek ve esnek olmayan şekilde ikiye ayrılabilir. Esnek olmayan yöntemde, önceden karar verilmiş sorular belirli bir sonuç elde etmek amacıyla bir düzen içinde kullanıcıya yöneltilir. Böylece belirli yüzdesel değerlere ulaşılır ve bunlar birbirleriyle karşılaştırılabilir. Esnek yöntemde ise, belirli konular olmakla birlikte, bir düzen yoktur ve kullanıcı soruları cevaplayıp cevaplamamakta, kendi görüşünü dile getirmekte tamamen serbest bırakılır (Dumas, Redish, 1993).

- **Anket ile Kullanıcı Görüşü Toplama:** Anket ile görüş toplarken mutlaka net ve kolay anlaşılabilir sorular kullanılmalıdır. Kullanıcının anketi cevaplarken belirli bir mesafede olacağı göz önüne alınmalıdır. Örneğin posta ile yapılan bir ankette, kullanıcı anlamadığı noktaları sorabilme şansına sahip değildir. Bu yüzden en azından bir adet pilot soru seçilmeli ve önceden denenmelidir. Genelde bu yöntemde kullanılan anketlerde, açık ve kapalı olmak üzere iki soru tipi bulunur.

Kapalı sorularda, kullanıcı kendisine verilen seçeneklerden durumuna uygun olanını seçer. Bu soru tipi için çeşitli skalalar mevcuttur. Basit bir kapalı soruda evet / hayır şeklindeki cevap tipi yeterliyken, daha karışık bir soruda 1'den 5'e kadar numaralandırılan çeşitli durumları içeren bir skala kullanılabilir. Açık soru tipinde ise, kullanıcıdan kendi düşüncesini açıkça yazması istenir. Bu anketlerin sonuçları çeşitli istatistik programlarının yardımı ile verilere dönüştürülebilir (Dumas, Redish, 1993).

- **Deneyler ve Ölçümler:** Buradaki deneyler ve ölçümler laboratuvar ortamında gerçekleştirilir. Ortaya bir hipotez konur ve bunu kanıtlamaya yönelik deneyler ve ölçümler yapılır. Yapılan deneylerin ve ölçümlerin sonuçlarının kabul edilebilir olması için, laboratuvar ortamındaki bütün değişkenler dikkatlice kontrol edilmelidir. Bu yüzden en zor değerlendirme yöntemlerinden biridir. Deneyin amacının ne olduğu, nelerin ölçüleceği, nelerin sabit tutulacağı gibi konular üzerinde mutlaka önceden düşünülmelidir (Laurel, 1990).
- **Anlama Dayalı Değerlendirme:** Bu değerlendirme yönteminde, tasarımcı kullanıcının kendi doğal ortamında ürünü nasıl kullandığını ve bu ürünü yaşantısıyla nasıl ilişkilendirdiğini anlamaya çalışır. Gözlemci, bulunduğu ortamda kimseye hissettirmeden ve informal metotları kullanarak bilgi toplar. Bu yöntem yaygın olarak kullanılan yöntemlerden birisidir. Bu yöntemi kullanmak isteyen bir gözlemcinin uzun süreler gözlemlediği insan veya insanlar ile beraber yaşaması gerekebilir (Laurel, 1990).
- **Tahmine Dayalı Değerlendirme:** Bu yöntemin amacı, kullanıcı olmadan ürünün problemlerini tahmin edebilmektir. Deneyimli eleştirmenler ürünü kontrol ederler ve sanki bir kullanıcıymışçasına problemlerini tahmin etmeye çalışırlar (Laurel, 1990).

Sonuç olarak, günümüzde, ihtiyaçları karşılayan, yeterli bir arayüz tasarımı bilgisayar sistemlerinin verimliliğini arttıran en önemli faktörlerden biridir. Mimarlık mesleği açısından ele alındığında ise, mimari yazılımların arayüzünün tasarımı, kullanılabilirlik açısından üzerinde çalışılması gereken bir konu olarak karşımıza çıkmaktadır.

4. MİMARLIKTA BİLGİSAYAR KULLANIMI

Bu bölümde, mimari yazılımların nasıl daha verimli kullanılabileceği sorusuna cevap aramak için, yazılım dünyasının tarihçesi, mimari yazılımlar ve diğer BDT (Bilgisayar Destekli Tasarım) yazılımları hakkında inceleme yapılmıştır.

4.1 Bilgisayar Destekli Tasarımın Gelişme Süreci

“Bilgisayar Destekli Tasarım /BDT” (Computer Aided Design and Drafting /CADD), Mimarlık mesleğinin altında ayrı bir uzmanlık alanıdır. Bugün, BDT(CAD) programların mimarlık eğitiminin ilk aşamasından başlayarak mimarlık mesleğinin içindedir (21).

BDT ile ilgili ilk çalışmalar 1950’lerde başlamıştır. Ancak bu çalışmalar mimarlık ve tasarım ile ilgili olmayıp doğrudan imalata yöneliktir.

İlk grafik sistemi, 1950’li yılların ortalarında, MIT Lincoln Laboratuvarı tarafından, Amerikan Hava Kuvvetleri için geliştirilen hava savunma sistemidir. SAGE (Semi Automatic Ground Environment) adı verilen bu sistem, radar bilgilerini ve gerekli bilgileri CTR monitörü yardımı ile göstermektedir.

1960 yılında, Ivan Sutherland, MIT Lincoln Laboratuvarında üretilen TX-2 adlı bilgisayarı kullanarak, SKETCHPAD adı verilen projeyi ortaya çıkarmıştır. Bu proje BDT endüstrisinin ilk önemli gelişmesi olarak kabul edilmektedir (21).

Buna benzer çalışmalar İTEK ve General Motors tarafından da geliştirilmiştir. İTEK tarafından geliştirilen “The Electronic Drafting Machine” vektörel bir monitöre (Vector-Refresh Display) ve geniş bir belleğe sahiptir. Komutlar bir ışıklı elektronik kalem yardımı ile girilmektedir. 1960 yılında “Mc Donnell Douglas Automation Company” (Mc Auto) kurulmuştur. Bu firma gelecekte BDT’nin gelişmesinde önemli role sahip olacaktır.

İlk BDT tasarım programları, çizgileri önce iki boyutlu, sonra üç boyutlu olarak gösteren basit bir algoritmaya sahiptirler. İlk tasarım programları ile ilgili çalışmalar Carnegie-Mellon Üniversitesi’nden Prof. Charles Eastman tarafından başlatılmıştır. Eastman tarafından geliştirilen “Building Description System” , yüzlerce mimari elemanı barındıran ve bunların bir tasarım fikrini oluşturmak için bir araya getirilebildiği bir kütüphanedir (21).

1965 yılında, Donald Welbourn ve Cambridge Üniversitesi'nden Prof T. F. Baker, Bilimsel Araştırma Konseyi'nin kendilerine verdiği 65.000\$ ile, BDT üzerinde çalışmalara başlamışlardır. Çalışmaları sırasında PDP11 grafik bilgisayarı kullanmışlardır ve bir çalışma grubu oluşturmuşlardır. Bu çalışma grubu içinde bulunan ve iki silindirin kesişme noktalarının nasıl tanımlanması gerektiği konusunda çalışan A. R. Forrest, ilk araştırma öğrencisi olmuştur.

Donald Welbourn, çalışmalarına üç boyutlu şekillerin modellenmesi konusunda devam etmiştir, fakat büyük bir maddi problem ile karşı karşıya kalmıştır. İlk olarak Ford daha sonra Volkswagen ve Daimler Benz firmaları Donald Welbourn'un çalışmalarına maddi destekte bulunmuşlardır.

1968'de David Evans ve Ivan Sutherland, Evans ve Sutherland firmasını kurmuşlardır. 1969'da ise Computer Vision ve Applicon firmaları kurulmuştur (21).

1970'lerde BDT sistemleri ile ilgili çalışmalar yapan pek çok firma kurulmuştur. Bunlardan "Applicon" diğerlerinde farklı olarak daha çok araştırmaya yönelik bir firmadır. Colma ve Auto-Trol Technology firmaları ise üretime yönelik olarak çalışan firmalardır. Bunların dışında General Motors, Ford, Chrysler ve Lockheed gibi bazı önemli otomobil ve havacılık mühendisliği firmaları " Bilgisayar Destekli Çizim ve Üretim /BDÇÜ (Computer Augmented Drafting and Manufacturing /CADAM) konusu üzerinde çalışmaktadırlar.

1975 yılında, Avions Marcel Dessault (AMD), firması BDÇÜ yazılımlarının lisans hakkını Lockheed firmasından satın alarak, en büyük CADAM üreticisi firma haline gelmiştir (21).

İlk katı modelleme yazılımları, 1970'li yılların sonunda üretilmeye başlanmışlardır. Bu programlar, küre, silindir, prizma gibi temel geometrik öğeleri alıp bunları birleştirerek çeşitli modeller oluşturabilmektedirler.

1977 yılında, Avions Marcel Dessault (AMD), üç boyutlu interaktif bir program olan CATIA (Computer Aided Three-Dimensional Interactive Application) programının prototipini geliştirmiştir. CATIA, BDÇÜ yazılımları konusunda bir dönüm noktası olmuştur. 1982 yılında CATIA'nın 1. versiyonu piyasaya sürülmüştür (21).

1982'de, maliyeti daha az, becerisi daha çok olan masaüstü bilgisayarları üretilmeye başlanmıştır. Bu önemli bir dönüm noktasıdır ve böylece teknoloji geleneksel yöntemler ile yarışmaya başlamıştır. Masaüstü bilgisayarların üretilmeye başlanması ile birlikte BDT programlarının kullanımı da yaygınlaşmaya başlamıştır. 3

Yukarıda bahsedildiği gibi, 1950'li yıllarda 1980'li yıllara kadar BDT programları, tasarımdan çok imalata yönelik olarak geliştirilmişlerdir. 1980'li yıllarla beraber, mimari tasarıma yönelik BDT programlarının geliştirilmesine başlanmıştır.

1982'de Autodesk firması, 16 kişi tarafında California'da kurulmuştur. Firmanın kuruluş amacı, kişisel bilgisayarlarda çalışabilecek ve 1000\$'a satılabilecek CAD programı üretmektir. 3

1982'de, Autodesk'ten sonra "CADPlan" adlı programı üreten P-CAD firması kurulmuştur. Bu ürün daha sonra CalComp firması tarafından satın alınmış ve CADVANCE olarak yeniden adlandırılmıştır.

1982 yılının Kasım ayında, Las Vegas'taki COMDEX fuarında, kişisel bilgisayarlar ile çalışan ilk BDT programı tanıtılmıştır. Bu "AutoCAD" programının ilk sürümüdür ve aynı yılın Aralık ayında satışa sunulmuştur.

1983 yılının Nisan Ayında AutoCAD 1. 2, Ağustos ayında 1. 3 ve ekim ayında 1. 4 piyasaya sürülmüştür.

Bu arada 1984 yılında, Gobor Major adlı bir Macar fizikçi, ülkesine gizlice iki adet Macintosh bilgisayar sokmayı başarmıştır. Macaristan'ın o zamanki komünist rejimi, kişisel bilgisayar sahibi olmayı yasaklamıştır. Bu şartlar altında, Bajor, Tomas Hajas adlı bir genç ile birlikte, Pascal programlama dilini kullanarak, Macintosh'da çalışabilen üç boyutlu bir BDT programı tasarlamayı başarmıştır (21).

AutoCAD, 1984 yılında, ilk eğitim merkezini kurmuştur. Aynı yılın Ekim ayında AutoCAD Versiyon 2 pek çok yeni özellikleri ile piyasaya sürülmüştür. "DXF" dosyası alma ve verme komutları, bilgi komutları, eleman yakalama, görünüş kaydetme, izometrik perspektif alma ve simge özellikleri bu versiyona eklenen özelliklerden bazılarıdır.

1985 yılında MicroStation firması, orijinal ismi "PseudoStation" olan ve Bentley Systems firması tarafından üretilen BDT yazılımını "MicroStation" adı altında yeniden piyasaya sürmüştür. MicroStation programı IGDS dosyalarını düzeltebilme yeteneğine sahiptir. Intergraph firması, Bentley Systems'in % 50'sini aldıktan sonra, MicroStation'un yeni versiyonunda IGDS dosyalarını DGN olarak yeniden adlandırmıştır (21).

1985 yılında, Autodesk'in satışları 26 milyon doların üzerine çıkmıştır. Aynı yıl Diehl Graphsoft firması kurulmuştur ve MiniCAD programının ilk versiyonu piyasaya sürülmüştür. MiniCAD, Macintosh üzerinde çalışan programlar içinde en çok satanı olmuştur (21).

1985 yılının Mayıs ayında AutoCAD 2.1 piyasaya sürülmüştür. Bu sürümde çoklu çizgi (polyline) komutu ve üç boyutlu çalışma özellikleri eklenmiştir.

1986 yılının Ocak ayında, AutoCAD 2.18 satışa sunulmuştur. Bu sürüm diğerlerinden farklı olarak AutoLISP programlama dilini içermektedir. Aynı yıl, AutoCAD PC World dergisinin verdiği “En iyi BDT ürünü” ödülünü kazanmış ve dünya çapında 50.000 adet satılmayı başarmıştır. 1986 yılının Haziran ayında AutoCAD 2.5 gelişmiş düzeltme ve ölçülendirme komutları ile satışa sunulmuştur.

BDT yazılım endüstrisindeki gelişmeler ve bu yazılımların kullanıcıların hızla çoğalması, bu endüstriden ve yazılımlardan haberler verecek bir yazılı medya ihtiyacını doğurmuştur. Bunun üzerine “CADANCE” dergisi yayımlanmaya başlanmıştır ve kısa zamanda bütün dünyada ilgi gören bir dergi olmuştur.

1987 yılının Nisan ayında AutoCAD 2.6 piyasaya sürülmüştür ve ilk kez AutoCAD sertifikası kursları İngiltere’de verilmeye başlanmıştır. Aynı yılın Eylül ayında AutoCAD’in 80x87 matematik işlemci gerektiren ilk versiyonu olan AutoCAD 9 satışa çıkarılmıştır. Sürüm 9 diğerlerinden farklı olarak, menü çubuğu, çekme menüleri, simge çubukları ve diyalog kutuları içeren, gelişmiş bir arayüze sahiptir. Sürüm 9 ile gelen diğer bir önemli gelişme, ADS’nin (AutoCAD Development System) eklenmesidir. ADS, AutoLISP yerine C kodlarının kullanılmasına izin vermektedir.

1988 yılının Ekim ayında AutoCAD 10 satışa sunulmuştur. Aynı yıl, Autodesk, AutoCAD için yardımcı bir yazılım olan AutoSolid’i piyasaya sürmüştür. 1989 yılında Autodesk “Generic Software” firmasını ve bu firmanın ürettiği “Generic CADD” programını satın almıştır. Diğer taraftan Amerika ve Kanada’da ArchiCAD ürünlerini satmak ve desteklemek amacıyla Grahisoft US kurulmuştur.

1990 yılında Bentley Systems firması “MicroStation” programının satışının 100.000 adete ulaştığını kamuoyuna duyurmuştur. Bu arada Autodesk “Animator-Pro” adlı DOS altında çalışan bir yazılım olan 2 boyutlu resim ve animasyon programını piyasaya sürmüştür. Aynı yılın Ekim ayında AutoCAD 11 satışa sunulmuştur. Diğer versiyonlardan farklı olarak, bir ağ üzerinden kullanılabilir ve pek çok yeniliklere sahiptir. Kağıt düzlemi ilk defa bu sürümde kullanılmıştır. Bu yılın sonunda Autodesk’in ürettiği AutoCAD, Generic CAD ve Auto Sketch programları dünyada en çok satılan BDT yazılımları olmuştur. AutoCAD “Byte Magazine” dergisinin “En iyi BDT ürünü” ödülünü kazanmıştır.

1991 yılında, Microsoft Open GL'i geliştirmeye başlamıştır. Open GL, üç boyutlu grafikler üretmek için kullanılan ve çizgi, nokta, poligon gibi çeşitli biçimleri çizmek için yaklaşık 120 komut içeren bir API işlemci yazılım arayüzüdür. Open GL Silicon Graphics tarafından geliştirilmiş olup, üç boyutlu renkli grafik programlama ve render için bir standarttır.

1994 yılında, pek çok programın yeni versiyonları piyasaya sürülmüştür. Autodesk 3D Studio Versiyon 4'ü ve AutoCAD Sürüm 13'ü çıkarmıştır. MicroStation Versiyon 5 ve MiniCAD Versiyon 5 piyasaya çıkarılmıştır. CINEMA 4D Versiyon 2. 0 ve bir peyzaj programı olan Site Designer 1 piyasaya sürülmüştür. "Numerical Software" firması "Visual CADD" programını satışa çıkarmıştır. Bu ürün bir yıl sonra "Corel Corporation" tarafından satın alınmıştır. Bu yılın sonunda AutoCAD 1 milyon adet ile dünyanın en çok satan BDT programı olmuştur. AutoCAD'in ardından 180.000 adet ile CadKey ve 155.000 adet ile MicroStation gelmektedir (21).

1995 yılında MicroStation programına katı modelleme özelliği eklenmiştir. Bu arada CINEMA 4D Versiyon 3. 0 piyasaya sürülmüştür. Autodesk ilk resmi web sitesini (www.autodesk.com) kurmuştur. Bu yılın sonunda AutoCAD en çok satılan BDT yazılımı olma özelliğini korumaktadır. Aynı yıl, Autodesk 3D Studio'nun NT platformunda çalışan versiyonu olan 3D Studio MAX'ı piyasaya sürmüştür. Bu arada, Bentley firması MicroStation 95'i piyasaya çıkarmıştır. Bu ürün "Basic" dili programlama araçları ve ODBC desteği içermektedir. Almanya'da ise Felix CAD firması "Felix CAD Versiyon 2" adlı yazılımı üretmiştir.

1995 yılının sonunda, Autodesk bir parametrik katı modelleme aracı olan AutoCAD Designer Sürüm 1. 2'yi üretmiştir. Bu programda yapılan modeller 3D Studio'ya aktarılabilir. "Mechanical Desktop'ın" ilk versiyonu da 1995 yılının sonunda satışa sunulmuştur. Mechanical Desktop AutoCAD Designer 3, Assembly Modeler 1, Auto Surf 3. 0, Auto Vision ve Part Spec Library'yi içeren bir sistemdir. Windows 95 ve NT altında çalışabilmektedir.

1996 yılının Ağustos ayında, Autodesk Mechanical Desktop Versiyon 1. 1'i satışa çıkarmıştır. Bundan kısa bir süre sonra, AutoCAD LT 95'i piyasaya sürmüştür. Aynı yılın Eylül ayında CINEMA 4D Versiyon 4 satışa çıkarılmış ve Windows, Alpha NT ve Mac platformlarında çalışabilmesinden dolayı büyük başarı kazanmıştır (21).

Aynı yıl Lightscape, Lightwave, Renderise Live adlı render ve animasyon programları üretilmiştir. Corel firması Corel Visual CADD 2 ve CorelCAD programlarını satışa sunmuştur. Felix CAD'in AutoCAD ile uyumlu 3. versiyonu piyasaya çıkarılmıştır. Graphisoft, Windows altında çalışan MiniCAD 6 programını, Bentley ise MicroStation Versiyon 4'ü üretmiştir.

1997 yılında, Autodesk AutoCAD LT 97, 3D Studio Max 2 ve bu programın mimari tasarım için hazırlanan hali olan 3D Studio Viz, AutoCAD modülleri için bir animasyon programı olan Autodesk Walk Through 1.1 programlarını piyasaya çıkarmıştır. İnternet üzerinden çizimlere ulaşmayı, görmeyi ve paylaşmayı sağlayan AutoCAD 13 İnternet Publishing Kit'i satışa sunmuştur. Bu yazılım, çizimleri DWF (Drawing Web Format) formatında sıkıştırıp İnternet'e koymaya yaramaktadır.

Aynı yıl, Bentley firması MicroStation Modeler 5.5'i piyasaya sürmüştür. Yine aynı yılın Haziran ayında Dessault Systems ve Solid Works firmaları, Windows altında çalışabilen mekanik tasarım yazılımı olan Solid Works'ü piyasaya çıkarmıştır. Bu ürün mekanik tasarım alanında oldukça başarı kazanmıştır (21).

1997 yılında, pek çok iki ve üç boyutlu yenilikler içeren AutoCAD 14 ve Mechanical Desktop 2 satışa çıkarılmıştır. Bu yılın sonunda AutoCAD'in satışı 1,5 milyon adet üzerine çıkmıştır. Form Z, MiniCAD Versiyon 7, SolidWorks 97, CINEMA 4D XL Versiyon 5. 0 programları piyasaya sürülmüştür.

1998 yılında, Autodesk, Architectural Desktop 1 ve 3D Studio MAX 2.5'i satışa çıkarmıştır. Aynı yıl SolidWorks 98, Lightwave 3D Versiyon 5.6 ve Visual Lisp programları da satışa çıkmıştır.

1999 yılında, MiniCAD programının yerine VectorWorks programı üretilmeye başlanmıştır. Graphisoft ayrıca VectorWorks'ü destekleyen ve foto-gerçekçi renderler alabilen Render Works'ü üretmiştir. Autodesk'in ürettiği 3D Studio MAX, render ve animasyon programlarının içinde en çok satanı olmuştur. 1999 itibari ile 100. 000 adet 3D Studio MAX satılmıştır.

2000 yılının Ocak ayında Nemetschek Kuzey Amerika'da VectorWorks Industry Series'in ilk ürünü olan "VectorWorks ARCHITECT'i" tanıtmıştır. Diğer program olan "VectorWorks SPOTLIGHT" ile "ARCHITECT" daha sonra birleştirilmiş ve "VectorWorks LANDMARK" adında tek bir program olarak satışa sunulmuştur.

Şubat ayının sonunda, Autodesk AutoCAD 2000'i Internet üzerinden satmaya başlamıştır. Bu sırada Nemetschek Almanya Diehl Graphsoft'u satın almıştır. Mart ayında da IntelliCAD ve LightWork Design programlarının yeni versiyonları çıkmıştır.

Haziran ayında Bentley Systems firması, mimarlar ve tasarımcıların yarattıkları büyük modeller için foto-gerçekçi ışıklandırma çözümleri içeren yeni bir teknoloji ürettiklerini açıklamıştır. "Particle Tracing" (Işın İzleme) adlı bu yeni ve kullanımı kolay teknoloji oldukça küçük bir bellek kullanmasına rağmen, çok gerçekçi ışıklandırma efektleri oluşturabilmektedir. (21).

Aynı yıl, Mechanical Desktop 5, 400'den fazla yeni ve geliştirilmiş özelliği ile piyasaya sürülmüştür. "Caligari" firması "True Space" adlı bir modelleme ve animasyon programını üretmiştir. Bu arada pahalı programlara seçenek olarak Chief Architect 7.0 adlı Windows tabanlı bir yazılım satışa sunulmuştur. Bu program AutoCAD 2000'in üçte bir fiyatına satılmaktadır.

4.2 Mimari Tasarımda Bilgisayar Kullanımı

Sayısal teknoloji, tasarım kültürünü, 1980'li yıllardan itibaren, önemli ölçüde etkilemiş ve değiştirmiştir. Günümüzde mimarlık öğrencileri, "Sayısal Dil" (Digital Language) olarak adlandırabilecek yeni bir ortak mimari dile bir ölçüde sahip olarak mezun olmaktadır (Laiserin, 2000). Sayısal teknoloji ve bilgisayar mimarlık mesleğine oldukça büyük yenilikler getirirse de, mimarların tasarım yaklaşımları hala aynı şekilde süre gelmektedir (Novitski, 1999).

Bilgisayar ve mimari yazılımlar tasarım süreci içinde, tasarımcılara yeterli ölçüde yardımcı olamamaktadır. Mimar için gerekli olan, fikirlerini kısa sürede bilgisayar ortamında modelleyebileceği bir yazılımdır. Günümüzde kullanılan yazılımlar ise, çalışma prensiplerinin mimari düşünce şekline uygun olmamasından dolayı yetersiz kalmaktadır (Laiserin, 2000).

Mimari tasarımda bilgisayar kullanımında karşılaşılan en büyük güçlük, gerekli verilerin bilgisayara girilmesidir. Mimar, eskiz kağıdına çizgi olarak çizdiği verileri bilgisayara tanımlarken bir çok teknik problem ile karşılaşmaktadır. Karşılaşılan bu teknik problemler, eskiz çizimlerinin ölçekli olarak çizilmesi sırasında karşılaşılan problemler ile benzerlik taşımaktadır.

Bilgisayar için çizgi, iki noktayı birleştiren bir doğru parçasıdır. Bir mimar için ise çizgi bir aksı, duvarı veya başka bir mimari elemanı ifade etmektedir. Mimar, tasarım yaparken, bilgisayar için anlamsız çizgileri yorumlayıp üçüncü boyuta yani gerçekliğe çıkarma yeteneğine sahiptir. Bilgisayarın ise, çizgileri birleştirip yorumlayabilmesi için daha fazla veriye ihtiyacı vardır. Mimarlık öğrencilerine gerekli bilgilerin verilmesi gibi bilgisayara da gerekli bilgilerin verilmesi lazımdır. Gelecekte çok daha hızlı donanımlar ve gelişmiş yazılımlar ile mimari yazılımların önceden bilgilendirilmiş olarak gelmesi mümkün olabilir. Günümüzde kullanılan tüm yazılımlar mimari bilgilerin tasarım aşamasında bilgisayara girilmesini gerektirmektedir.

Mimari yazılımların, mimari tasarım sürecine en önemli katkısı model oluşturmaktır (Novitski, 1999). Bilgisayar modeli, simülasyon ve üretim için gerekli veri tabanını oluşturmaktadır. Tasarımları bilgisayar ortamında modellenmesi, çeşitli mimari analizlerin bilgisayar tarafından yapılmasını sağlamaktadır. Sayısal model hazırlanması, bilgisayarın tasarımcıya tasarım sürecinde yardımcı olmasını sağlamaktadır.

W. Mitchell, 17 Ocak 2000 tarihinde Yıldız Teknik Üniversitesi'nde verdiği konferansta, mimarlık ve bilgisayar ilişkisi üzerine bir sistem önermiştir. Mitchell, bu sistem ile, bilgisayarı tüketim malzemesi olmaktan çıkarıp, üretim mekanizmasına dönüştürmeyi hedeflemektedir. Buna göre Mitchell bilgisayar kullanımını dörde ayırmaktadır:

- Simülasyon aracı
- Hızlı modelleme aracı
- Ortak çalışma
- Sanal tasarım stüdyosu

4.2.1 Simülasyon Aracı

Sürekli tekrarlayan matematiksel işlemlerin bir makine tarafından yapılması amacıyla, 19.yy başında yapılan çalışmalar modern bilgisayarların icadına öncülük etmiştir. Bilgisayarların verilen işlemleri sınırsız olarak tekrarlayabilmesi, insan ömrünün yetmeyeceği hesaplamaların anında yapılabilmesini sağlamaktadır (Ifrah, 2000). Sonsuz işlem yapma kapasiteli bilgisayarlar, önceden tanımlanmış fonksiyonlar ile belirlenmiş yapay bir dünya barındırabilir. Bu yapay dünya içinde betimlenen her durum için bilgisayar sonsuz sayıda analiz yapabilir.

Bilgisayar simülasyonları sayesinde, sadece tanımlanmış sistemler ile analizler yapılabilir. Bilgisayar simülasyonu dünyayı betimlemek için yapılır (13). Ama simülasyonlar gerçek dünyayı tamamıyla betimleyememektedir. Çünkü gerçek dünya betimlenemeyecek kadar zengindir. Deneyimler sayısallaştırılmamaktadır (Walker, 1994).

Gerçek zamanlı simülasyonlar kolayca ulaşılabilen görsel, interaktif bir model sağlar. Bu model üzerinde alternatif çözümler kısa sürede geliştirilebilmektedir. Bu tip simülasyonlar maliyeti ve yapım süresini azaltmaktadır. Amerika Birleşik Devletleri Philadelphia kenti için yapılması planlanan bir beysbol stadı, görsel simülasyonlar ile analiz edildiğinde yerinin hatalı olduğu ortaya çıkmıştır (43).

Massachusetts Institute of Technology'de "Growltiger" adlı bir yazılım ile öğrenciler taşıyıcı sistemi analizleri yapmaktadırlar. Bu şekilde, öğrenciler, yaptıkları tasarımların eksiklikleri öğrenip, gerekli değişiklikleri yapmaktadırlar (Winograd, 1996).

Bilgisayar simülasyonları, sosyal araştırmalar için de kullanılmaktadır. Forrester "Sistem Dinamikleri" (Systems Dynamics) denilen bir metot geliştirmiştir. Bu metot, 1970'lerde, endüstrileşmenin getirdiği yüksek nüfus artışının getirdiği dengesizliği tanımlamak için kullanılmıştır (47).

Mimari alanda kullanılan simülasyon teknikleri şöyle sıralanabilir:

- **Mimari Analizler:**

- Kesit ve görünüş hazırlayıp yapı ile karşılaştırma
- Detayların proje ile uyumu
- Ses ve ışık gibi yapı fiziği analizleri
- Fonksiyon analizleri
- Yapı analizleri

- **Görsel Simülasyon**

- Render ve animasyon teknikleri ile fiziki çevre uyumu
- Sanal gerçeklik ile mekan içinde yaşama

- **Statik Analizler**

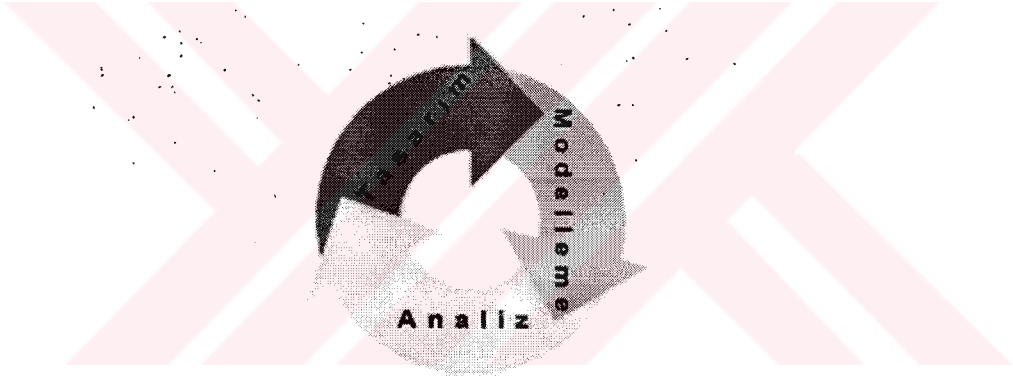
- Taşıyıcı sistem tasarımı
- Yük dayanımı hesapları
- Deprem testleri
- Mekanik testler
- Yapı yorulma testleri

- **Yönetim Sistemleri**

- Maliyet hesapları
- Proje uygulama metotları
- İş yönetimi

4.2.2 Modelleme Aracı

Bilgisayar modeli verilerden (data) oluşmaktadır. Bir modelin zenginliği içinde barındırdığı veri miktarı ile ölçülmektedir. Bir simülasyon aracı olan bilgisayar için model esastır. Bilgisayar ortamında model oluşturma uzun sürmekte ve zordur. Tasarım sürecinde oluşturulan model, simülasyon araçları ile analiz edilip, tasarıma etki edebilecek yeni girdiler sağlamada kullanılmaktadır. Tasarım süreci, modelleme ve analiz aşaması eş zamanlı yürütülmelidir (29). Bilgisayar modellerinin prototipleri, bilgisayar destekli üretim sistemleri ile üretilebilmektedir. Bu prototipler, simülasyon sonuçlarında olduğu gibi, tasarıma etki edebilecek yeni girdiler sağlayabilmektedir (Şekil 4.1).



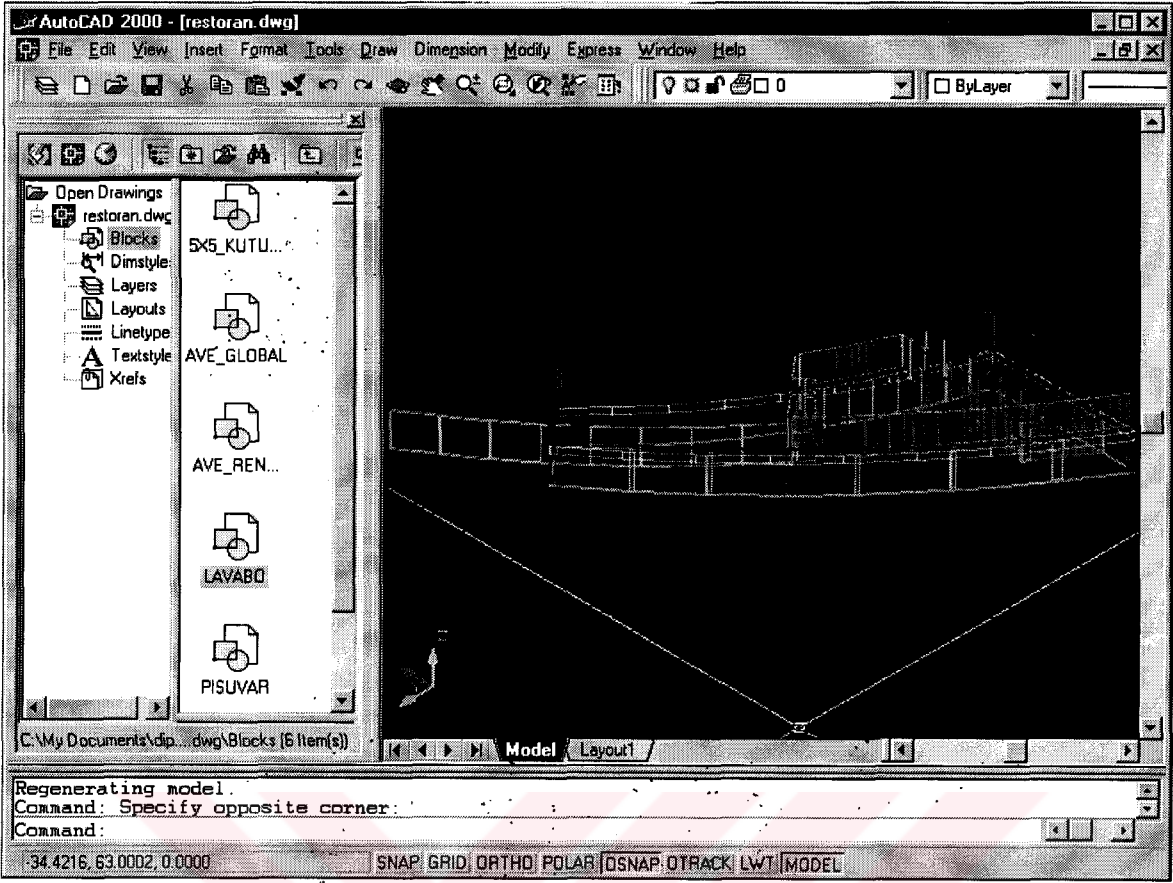
Şekil 4.1 Tasarım, modelleme ve analiz ilişkisi (29)

Modelleme süreci içerisinde mimari tasarımda kullanılan yazılımlar şöyle sıralanabilir:

4.2.2.1 Genel Modelleme Yazılımları

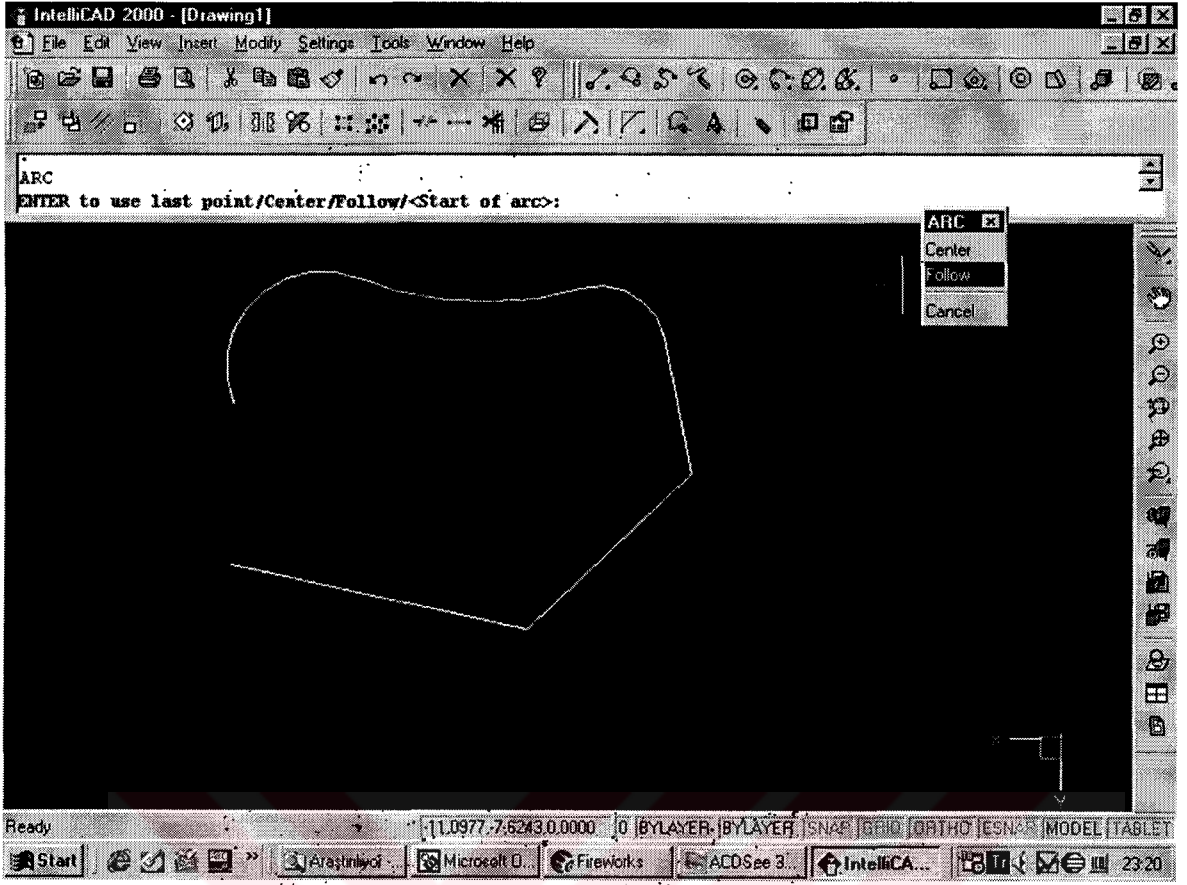
Temel geometrik formlar ile her türlü modelin yapılabilmesi için geliştirilen yazılımlardır. Çizgi, daire, kare, dikdörtgen gibi iki boyutlu geometrik formlar ile küp, dikdörtgen, küre, silindir gibi üç boyutlu geometrik formlar ile model oluşturulur. Modeli oluşturan biçimlerin sadece geometrik olarak tanımlanması, model üzerinde analiz yapmayı zorlaştırmaktadır. Yaygın olarak kullanılan genel modelleme yazılımlarının bazıları şunlardır:

- 1) AutoCAD: Autodesk tarafından geliştirilmektedir. İlk kez 1983 yılında piyasaya sürülmüştür. Donanım gücü zayıf olan sıradan bir kişisel bilgisayarda çalışabildiği ve performansı yüksek olduğu için kısa sürede yaygınlaşmıştır. 1986 yılında 50.000 adet satılarak en yaygın BDT yazılımı olmuştur (Walker, 1994). AutoCAD ile her türlü iki ve üç boyutlu modeller oluşturabilir. Bu modeller mimari bir proje veya uçak tasarımı olabilir. Modeller ölçülendirilebilir ve modeller üzerinde tarama yapılabilir, yazı yazılabilir. Katman ve kütüphane elemanı kullanıldığında modelleme süreci kısalmaktadır. DWG (Drawing File) ve DXF (Drawing Exchange File) adında iki dosya formatını endüstri standardı haline getirmiştir. AutoCAD'in parametrik özelliği bulunmamaktadır. Ama Autodesk tarafından geliştirilen parametrik yazılımlara temel oluşturmaktadır. Arayüzü DOS işletim sistemi için geliştirilmiş ve Windows işletim sistemine adapte edilmiştir. Sadece Windows işletim sistemi olan kişisel bilgisayarlarda çalışabilmektedir (4) (Şekil 4.2).



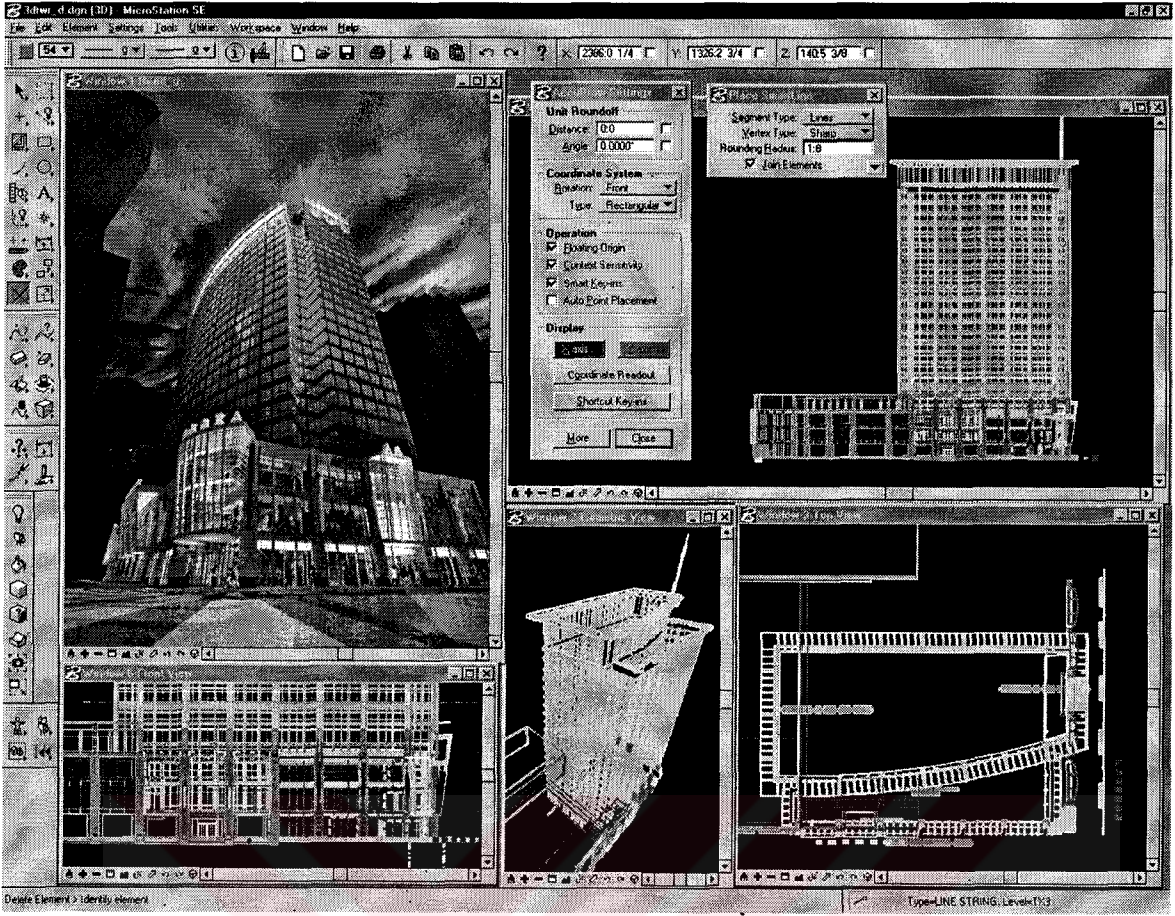
Şekil 4.2 Autodesk AutoCAD 2000 (4)

- 2) IntelliCad: BricsNet firması tarafından AutoCAD'e rakip olarak geliştirilmektedir. AutoCAD yazılımının tüm fonksiyonlarını yerine getirmektedir (7) (Şekil 4.3).



Şekil 4.3 BricsNet IntelliCAD (7)

- 3) MicroStation: Bentley firması tarafından geliştirilmektedir. AutoCAD'in yapısına benzemektedir. MicroStation yazılımının parametrik özelliği bulunmaktadır. Yazılımın arayüzü Windows işletim sistemine uygun olarak düzenlendiğinden, kullanımı daha kolaydır. Windows dışında işletim sistemleri için farklı sürümleri de vardır (5) (Şekil 4.4).

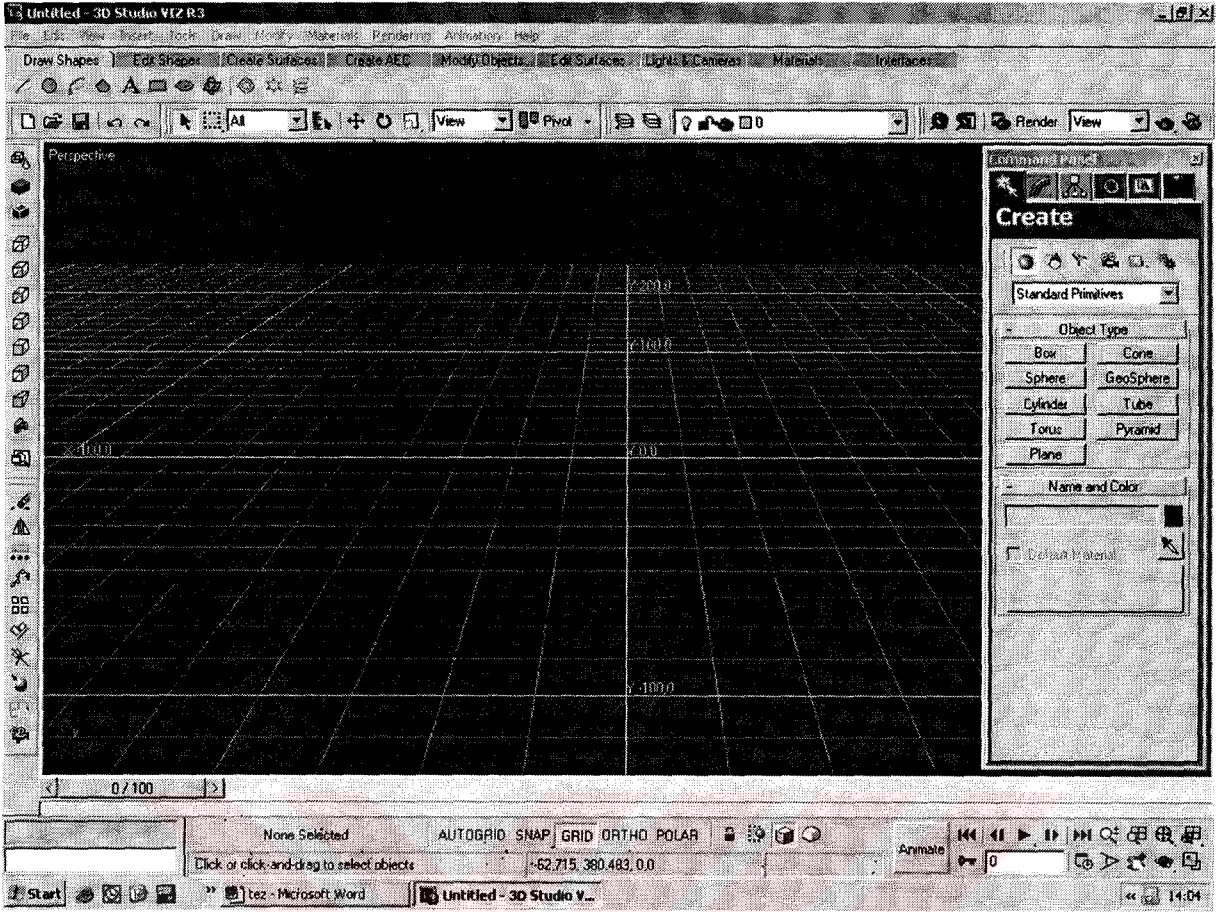


Şekil 4.4 Bentley MicroStation (5)

4.2.2.2 Görsel Modelleme Yazılımları

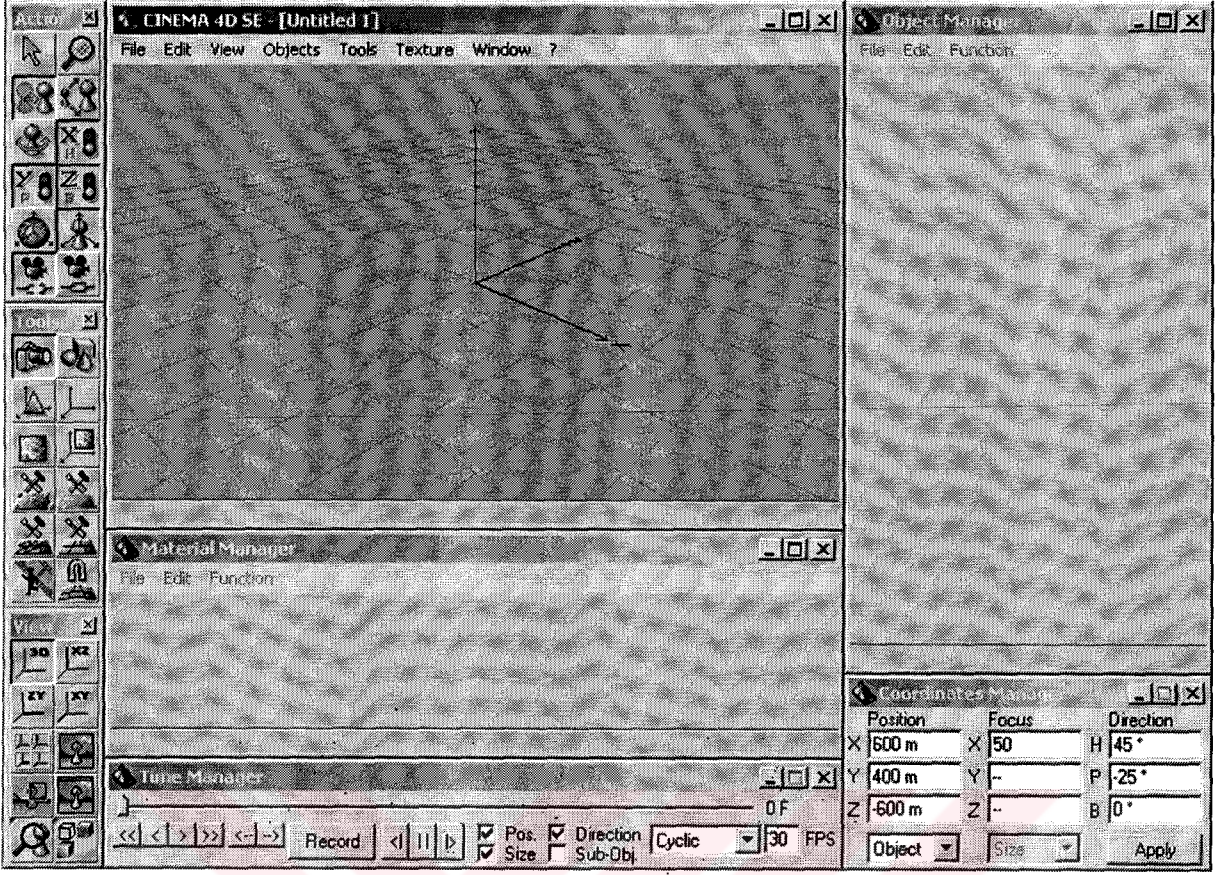
Bu tür yazılımlar simülasyonlar için üç boyutlu model oluşturmaktadırlar. Bu modeller temel geometrik biçimler ve eğrisel yüzeylerden oluşabilir. Modeller arasında parametrik bağlantılar olabilmektedir. Modellerin boyutlandırılması, diğer modelleme yazılımları ile karşılaştırıldığında, fazla hassas olamamaktadır. Yaygın olarak kullanılan görsel modelleme araçları şunlardır:

- 1) 3dStudioViz: Kinetix firması tarafından geliştirilmektedir. AutoCAD ile beraber çalışabilmektedir. DWG dosyalarına parametrik bağlantı kurulabilir. Gelişmiş kamera ve ışıklandırma araçları vardır. Modellere hareket verilebilir ve animasyon çekilebilir (26) (Şekil 4.5).



Şekil 4.5 Kinetix 3DStudioViz (26)

- 2) Cinema4D: Üç boyutlu modelleme ve sunum yazılımıdır. Gelişmiş hareket ve animasyon özellikleri vardır. Cinema4D için geliştirilmiş animasyona yönelik özel efektleri vardır (10) (Şekil 4.6).

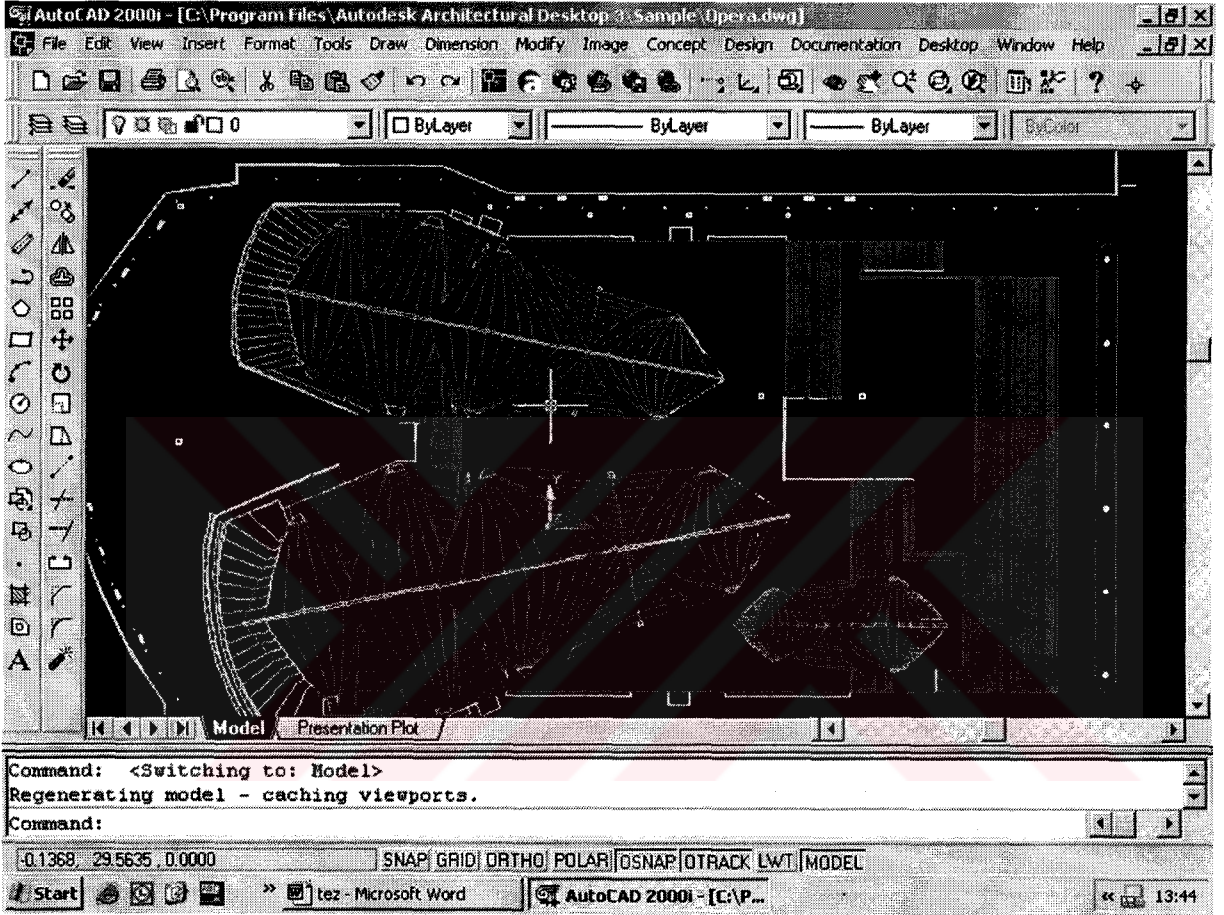


Şekil 4.6 Cinema4D (10)

4.2.2.3 Mimari Modelleme Yazılımları

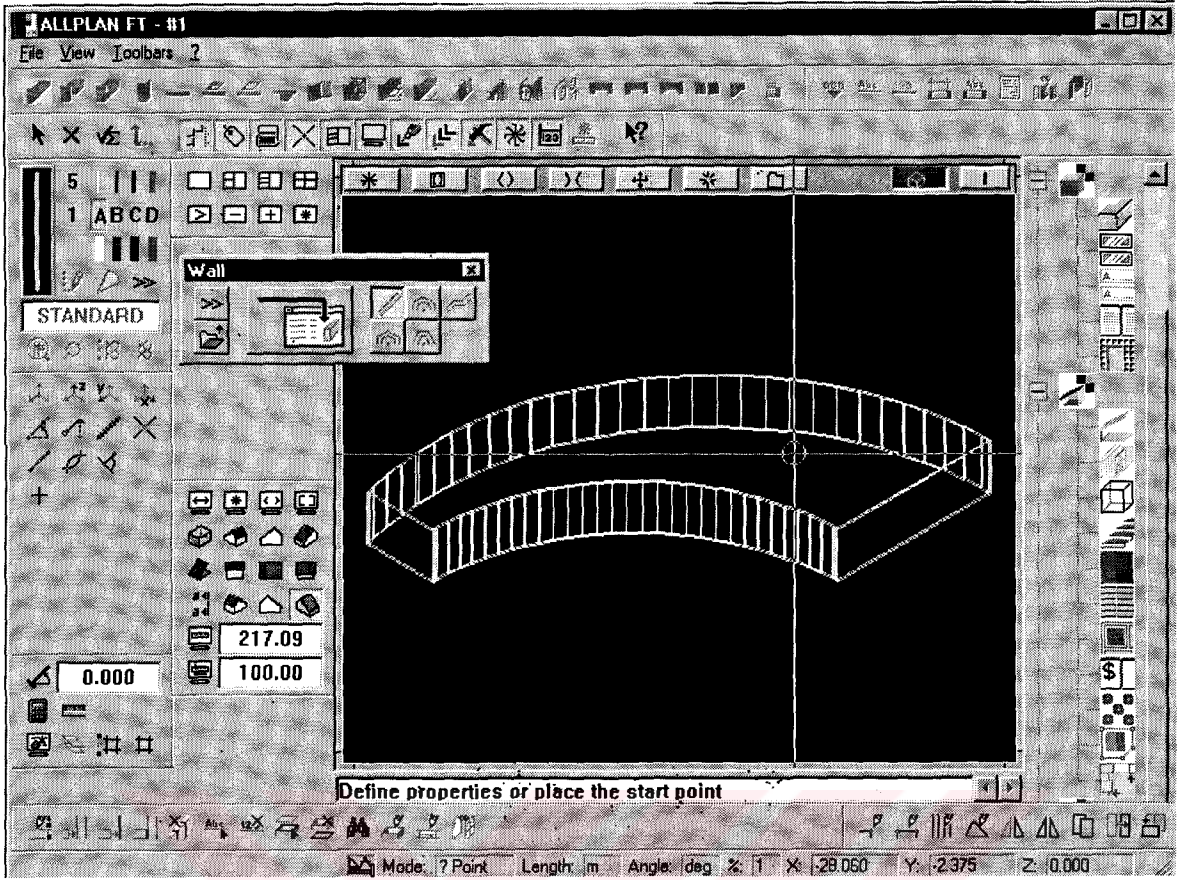
Temel geometrik formlar ve mimari elemanlar ile model hazırlanabilen yazılımlardır. Duvar, döşeme, pencere, kapı gibi temel mimari öğeler modele eklenerek proje hazırlanır. Mimari öğelerin model içinde kullanılması, bilgisayar simülasyonlarının daha kolay yapılmasını sağlar. Örneğin, mimari modelleme yazılımları ile kesit ve görünüş alınabilir ve üzerlerinde yapı elemanı boyutunda çalışma yapılabilir. Genel olarak kullanılan mimari yazılımlar şunlardır:

- 1) Architectural Desktop: Autodesk firması tarafından geliştirilmektedir. AutoCAD yazılımı üzerine kurulmuştur. Yazılımın arayüzü, DOS işletim sistemi için geliştirilen AutoCAD arayüzünün Windows işletim sistemi için geliştirilmiş halidir. Bu yüzden öğrenim ve kullanım problemleri olabilmektedir. Architectural Desktop'taki tüm mimari öğeler üç boyutlu olarak projeye eklenebilir. Kesit ve görünüş alınabilir ama bunların üzerlerinde çalışma esnek değildir (4) (Şekil 4.7).



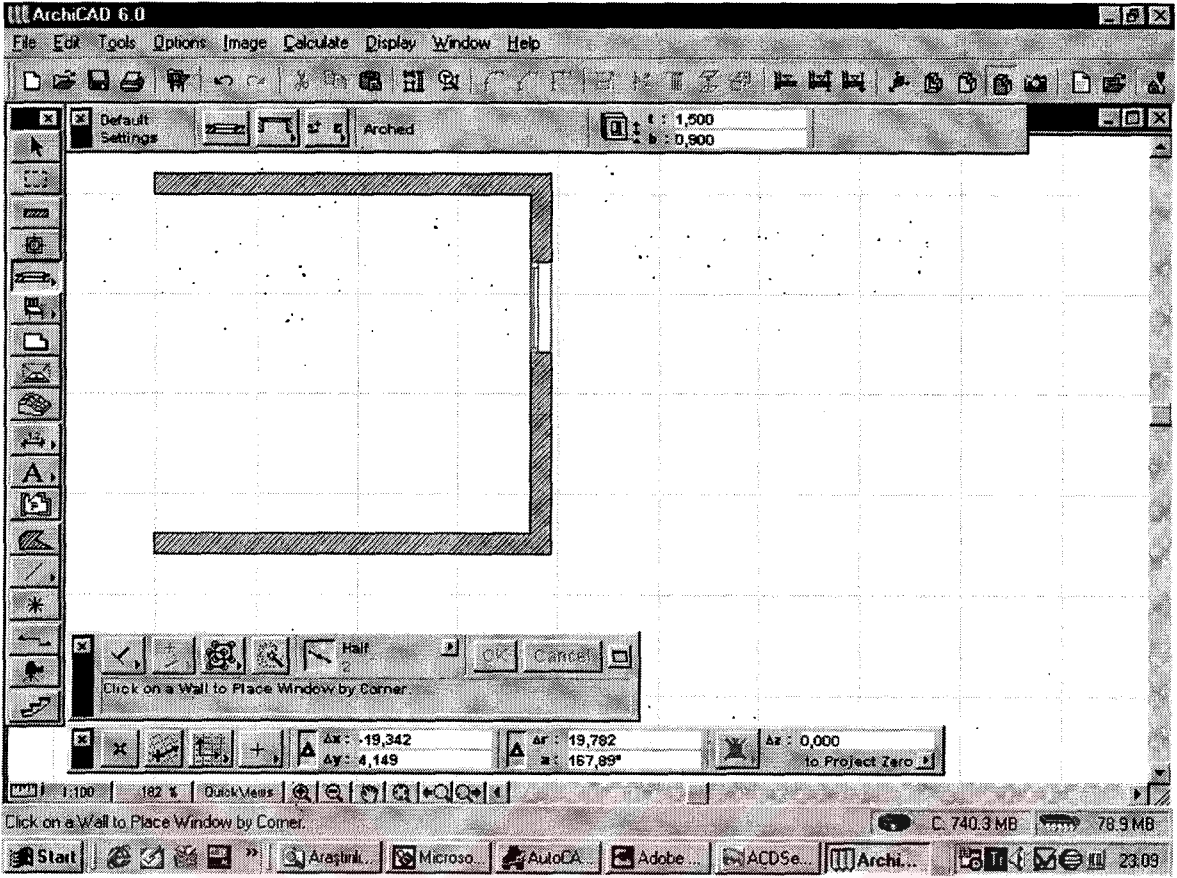
Şekil 4.7 Autodesk Architectural Desktop (4)

- 2) Allplan: Nemetschek tarafından geliştirilmektedir. Unix üzerinde geliştirilip Windows işletim sistemine uyarlanmıştır. Arayüzü Windows işletim sistemine uyumlu değildir. Öğrenim ve kullanım zorlukları vardır. Allplan ile üç boyutlu mimari elemanlar ile model hazırlanabilmektedir. Kesit ve görünüş üzerinde çalışılabilmektedir. Hakediş hesapları yapılabilmektedir. Türkçe versiyonu da bulunmaktadır (34) (Şekil 4.8).



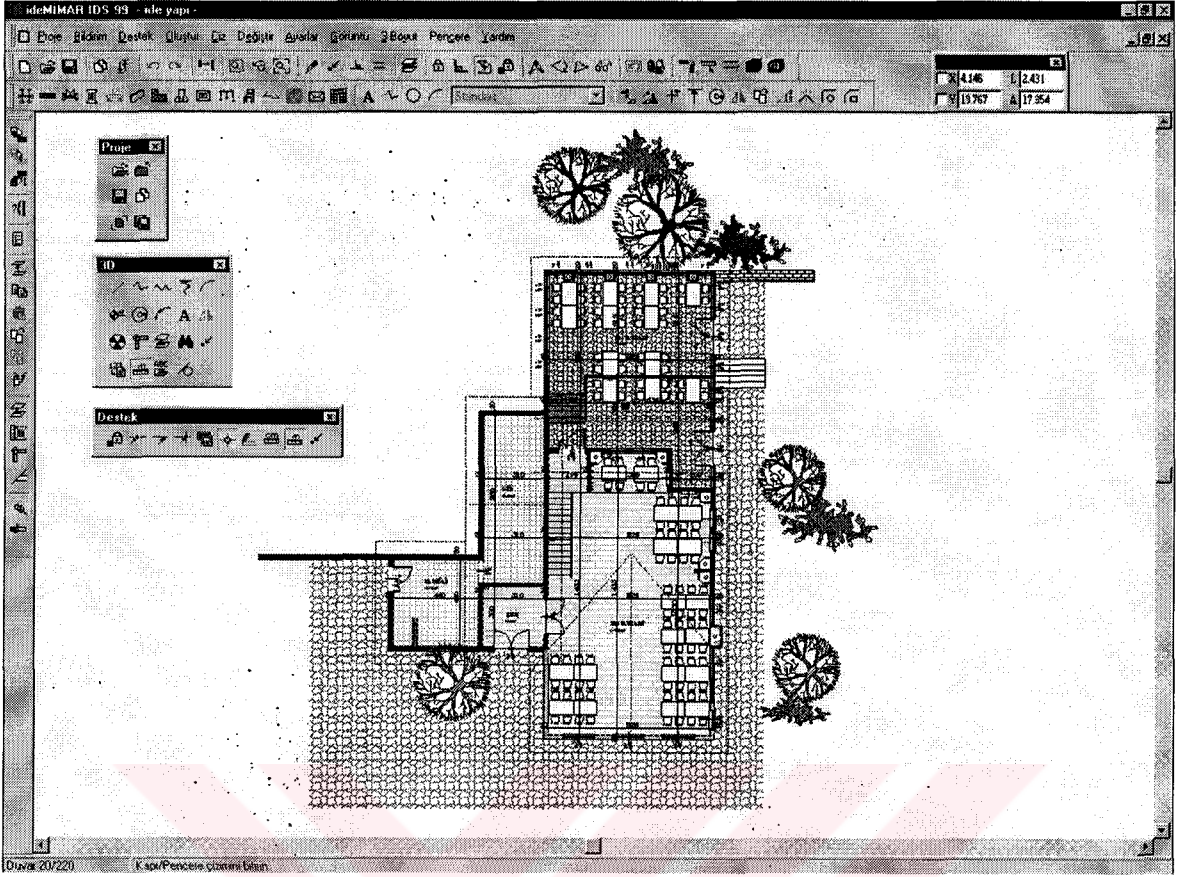
Şekil 4.8 Nemetschek Allplan (34)

- 3) ArchiCAD: Graphisoft tarafından geliştirilmektedir. Macintosh işletim sistemi için geliştirilmekte ve Windows işletim sistemine uyarlanmaktadır. Yazılımın arayüzü Macintosh işletim sistemi için geliştirildiğinden, Windows kullanıcılarında uyum problemi olabilmektedir. Kesit ve görünüş ile çalışılabilmekte, ışık analizleri yapılabilmektedir. Ayrıca hakediş hesapları yapılabilmektedir. Türkçe versiyonu da bulunmaktadır (14) (Şekil 4.9).



Şekil 4.9 Graphisoft ArchiCAD (14)

- 4) IdeCAD: İde Yapı tarafından Türkiye’de geliştirilmektedir. BDT’de kullanılan terimlerin Türkçe’de tam karşılığı olmamasından dolayı İngilizce geliştirilmektedir. İngilizce, Almanca ve Türkçe olarak satılmaktadır. Kesit ve görünüş alınabilmektedir. IdeStatik yazılımı ile beraber çalışabilmektedir. IdeCAD içindeki yapı elemanları statik bilgi içermektedir (23) (Şekil 4.10).



Şekil 4.10 İdeYapı İdeCAD (23)

4.2.2.4 Taşıyıcı Sistem Tasarımı Yazılımları

Statik özellikleri olan elemanlar ile taşıyıcı sistem tasarımı yapılan yazılımlardır. Taşıyıcı sistem tasarımında statik analizlerin yapılabilmesini sağlar. Ayrıca maliyet ve iş yönetim planları da hazırlanabilmektedir. Kullanılan bazı taşıyıcı sistem yazılımları şunlardır:

- 1) XSteel: Çelik yapı tasarımı için geliştirilmiştir. Endüstride üretilen çelik profilleri ile taşıyıcı sistem tasarımı yapılabilir ve statik analizleri hesaplanabilir (46).
- 2) Sap2000: Çelik ve betonarme yapı tasarımı için geliştirilmiştir. Tanımlanan her türlü statik eleman ile statik analiz yapılabilir (41).
- 3) İdeStatik: Betonarme yapı tasarımı için geliştirilmiştir. Statik yapı elemanları ile taşıyıcı sistem tasarımı yapılmaktadır (23).

4.2.2.5 İş Yönetimi Yazılımları

Maliyet analizleri ve iş programları ile iş yönetimi yapılan yazılımlardır. Genel olarak kullanılan yazılımlar şunlardır:

- Microsoft Project Management (31).
- Primavera (39).

4.2.3 Ortak Çalışma (Collaboration)

Günümüzde farklı üretim alanlarında bireyler ve organizasyonlar, bilgisayar sistemleri aracılığı ile iletişimde bulunup işbirliği yapabilmektedirler (Sarıyar, 2001). Bilgisayar destekli ortak çalışma, bireylerin bir grup olarak bir arada çalışmalarından anladıkları ile buna izin veren bilgisayar ağları teknolojileri, donanım, yazılım, servisleri ve tekniklerini birleştirir (Wilson, 1991). Bilgisayar destekli ortak çalışma, ortak çalışma amacındaki bireylerin bilgisayar destekli koordinasyon ile yürüttükleri etkinlik olarak tanımlanır (Baecker, 1993).

Bilgisayar Destekli Ortak Çalışma'nın gücü, katılımcıların aynı mekan-zaman dilimini paylaşmamasına karşın ortak amaca yönelik birlikte çalışabilme olanağı tanimasından kaynaklanmaktadır. Ortak çalışma süreci içerisinde, katılımcılar, buldukları yer ve zamana göre farklı iletişim gereksinimlerine yanıt getirebilmek için, elektronik ortamlar aracılığı ile ses, görüntü ve veri iletişimine girmektedirler (Sarıyar, 2001).

Mimari ortak çalışma için bilgisayar ortamında kullanılan yazılımlar şu şekilde gruplanabilir:

- Anında haberleşme yazılımları
- Yazı, ses ve görüntü ile haberleşme araçları
- Dosya transferi ve dosya paylaşımı

4.2.3.1 Anında Haberleşme Yazılımları:

Abone sistemi ile çalışan ve anında haberleşme imkanı sunan yazılımlardır. En yaygın kullanılan haberleşme yazılımları şunlardır:

- 1) ICQ: “Mirabilis” yazılım firması tarafından geliştirilmekte olan anında mesajlaşma yazılımıdır. Ücretsiz olarak kullanılmaktadır. En yaygın olarak kullanılan anında haberleşme yazılımıdır. Her abonenin bir numarası vardır ve bu numara kimlik yerine geçmektedir. Eklenti yazılımlar yardımıyla sesli veya görüntülü haberleşme yapılabilmektedir. Haberleşme için tarafların birbirlerini onaylaması gerekli değildir. Çevrimdışı abonelere mesaj yollanabilmektedir. Bir çok işletim sistemi ile kullanılabilir. Şirket içinde kullanmak için Intranet üzerinden çalışan bir versiyonu da vardır. İnternet üzerinde herhangi bir tarayıcı ile de kullanılabilir (32). Yıldız Teknik Üniversitesi'nin katıldığı MYS uzaktan erişimli projesinde de kullanılmıştır (Sarıyar, 2001).
- 2) AOL: “American Online” firmasının geliştirdiği ve Netcastor'a üye olan herkesin ücretsiz olarak kullanılabilen anında haberleşme yazılımıdır. Taraflar birbirlerini tanıladıktan sonra yazı veya sesli mesaj atabilmektedirler. Amerika'da yaygın olarak kullanılmaktadır (50).
- 3) Messenger: “Microsoft” firmasının ürettiği anında haberleşme yazılımıdır. 2000 yılından sonra çıkan Windows işletim sistemlerinde standart olarak kurulmaktadır. Microsoft'un “.NET” projesi çerçevesinde geliştirilmektedir. MSN'ye abone olarak bir “Pasaport” alan herkes ücretsiz olarak kullanabilmektedir (30).
- 4) Özel Konuşma Odaları: MIRC gibi yardımcı yazılımlar ile bağlanılır. Herkes konuşma odası açabilir. Bu odalara kimlerin girebileceği sınırlanabilir. Odalardaki üyeler ile dosya paylaşımı yapılabilir (44).

4.2.3.2 Yazı, Ses ve Görüntü ile Haberleşme Araçları:

Video konferans yapılması için kullanılırlar. Yıldız Teknik Üniversitesi'nin katıldığı MYS uzaktan erişimli projesinde bu tür yazılımlar kullanılmıştır. Sarıyar, kullanılan yazılımları şöyle sıralar:

- 1) NetMeeting: Messenger yazılımı beraber kullanılmaktadır. Bir “pasaport” alan herkes ücretsiz olarak kullanabilir. Konuşma odası, dosya paylaşım özelliği ve şekil çizmek için yazı tahtası vardır. Ayrıca bilgisayar ekranının tamamı veya bir uygulama paylaşımına açılabilir. Video konferans yapılabilir (31).

2) PictureTel: Picture Tel, profesyonel sesli ve görüntülü haberleşme araçları üreten bir kuruluştur. Yazılımın yanı sıra haberleşme için özel donanımlar da üretmektedir. Bu donanımlar sayesinde tüm haberleşmeler kaydedilebilir ve farklı medya ortamlarına aktarılabilir (Picturetel) (37).

4.2.3.3 Dosya Transferi ve Dosya Paylaşımı:

Internet veya LAN üzerinden dosya paylaşımı sağlanması için kullanılan ek yazılımlardır. En yaygın kullanılanları şunlardır:

- 1) AutoCAD External Reference: AutoCAD yazılımı için geliştirilmiştir. Tüm AutoCAD tabanlı yazılımlarda kullanılmaktadır. AutoCAD DWG dosyalarına referans vererek yeni çizimler eklenmesine yardımcı olur (4).
- 2) ArchiCAD TeamWork: Graphisoft Firmasının ArchiCAD yazılımı için geliştirdiği bir eklentidir. Hem ArchiCAD dosyalarına, hem de AutoCAD DWG dosyalarına referans verilebilir (14).
- 3) IdeCAD Takım Çalışması: İde Yapı Tarafından geliştirilmiştir. IdeCAD ile beraber kullanılmaktadır. IdeServer Yazılımı ile bir proje paylaşımına açılıp, kullanıcılara belirli haklar sunulmaktadır. Her kullanıcı kendine tanınan haklar çerçevesinde proje üzerinde çalışma yapabilmektedir (23).

4.2.4 Sanal Tasarım Stüdyosu (Virtual Design Studio)

Sayısal ortamlarda varolan sanallık kavramı, tasarım stüdyosu ile birleştirildiğinde, tasarım ürününü veya bilgi ve verinin sayısal ortamlarda paylaşıldığı fiziki sınırlardan bağımsız stüdyo anlamına gelmektedir (Sariyar, 2001).

Sanal tasarım stüdyoları, iletişim teknolojisini kullanarak bir araya gelmesi zor olan bireyleri bir araya getirerek iletişim ortamını zenginleştirebilirler (Wojtowics, Davidson, Nagakura, 1995). Sanal tasarım stüdyoları, geleneksel stüdyo paylaşımına destek bir iletişim aracı olarak kullanılabilirler gibi, çalışma süreci içerisinde bilgi paylaşımının yapılabildiği tek ortam da olabilirler.

Sanal tasarım stüdyolarına katılımcılar bir kullanıcı adı ve şifre ile bağlanırlar. Kullanıcılar dosya yükleyerek sunum yapabilir veya başka kullanıcıların sunumlarını izleyebilirler. Sanal tasarım stüdyolarında kullanıcılar bir bilgisayar ağının parçasıdırlar. Uzaktan erişimli tasarım stüdyosunda bilgisayar destekli ortak çalışma araçlarının kullanımı, katılımcıların iletişimini sağlar. Böylece ortak tasarım sürecinde üretim paylaşımı gerçekleştirilebilir.

2001/2002 öğretim yılında Yıldız Teknik Üniversitesi, Massachusetts Institute of Technology ve Saraybosna Üniversitesi ile beraber bir sanal tasarım stüdyosu kurmuştur. Bu stüdyoda katılımcılar, ArchNet adında özel bir Internet sitesini kullanarak ortak çalışma yapmışlardır. ArchNet kullanıcılara dosya yükleme ve indirme olanağı sunmuştur. Kullanıcılar yüklenmiş dosyalara yorum yapabilmiş ve özel konuşma odasını kullanarak görüşme yapmışlardır.

Sonuç olarak, mimarlıkta bilgisayar kullanımı çok çeşitli şekillerde karşımıza çıkmaktadır. Bunlar içinde en yaygın olanı, mimari yazılımlardır. Mimari tasarım için kullanılacak bir mimari yazılımın, bilgisayarın mimarlıktaki diğer kullanım şekillerini de içermesi (ortak çalışma, simülasyon vb.) konusunda çalışmalar yapılmaktadır.

5. BİR KAVRAMSAL MİMARİ YAZILIM ARAYÜZÜ ÖNERİSİ

Mimari tasarımda farklı amaçlar için üretilmiş birçok yazılım vardır. Mimari tasarımda bilgisayar kullanımı çeşitlilik göstermektedir. Bir mimari tasarım yazılımı her tasarım ihtiyacını karşılamak durumundadır. Tasarım ihtiyaçlarının tamamını kapsayan bir mimari yazılım yoktur. Bu amaçla tüm tasarım ihtiyaçlarını karşılayabilecek bir kavramsal yazılım arayüzü önerisi hazırlanmıştır.

Bilgisayar teknolojileri ve mimari tasarımda bilgisayar kullanımı birleştirilerek, kavramsal bir mimari yazılım arayüz önerisi yapılmıştır. Bu öneri, mimari yazılımların olumlu ve olumsuz yönleri incelenerek hazırlanmıştır. Bir ihtiyaç listesi hazırlanarak arayüz tasarımı yapılmıştır. Bu arayüz tasarımı bilgisayar ortamında modellenerek yazılım haline getirilmiştir.

5.1 İhtiyaç Listesi

- **Üç boyutlu tasarım:** Tasarımlar duvar, kapı, pencere, kolon gibi gerçek üç boyutlu mimari elemanlar ile yapılmalıdır.
- **Gelişmiş merdiven tasarımı:** Programın geniş merdiven kütüphanesi kullanılarak istenilen tipte merdiven tasarlanabilmelidir. Korkuluk, küpeşte ve basamak detayları otomatik olarak oluşturulmalıdır.
- **Kapı ve pencere tasarımı:** Programın kapı-pencere kütüphanesi içindeki mevcut tipler kullanılabilmesi gibi, bunların üzerinden değişiklikler yapılarak istenilen tipte kapı ve pencereler tasarlanabilmelidir.
- **Tonoz ve kubbe tasarımı:** İstenilen kesitte tonoz ve kubbe oluşturulabilmelidir.
- **Eskiz çizerek tasarım yapma:** Eskiz komutu ile çizim yapıldığında çizilen elemanlar üç boyutlu mimari elemanlar olarak tanımlanabilmelidir.
- **Mimari Tasarım:** Mimarlar eskiz ile başlayan tasarımlarını, hiç ödün vermeden bilgisayar ortamında modelleyebilmelidirler.
- **Üç boyutlu detay tasarımı:** Detaylar üç boyutlu tasarlanmalıdır. Bu şekilde projeye uygun gerçekçi çözümler üretilebilir.
- **Akıllı eleman mantığı:** Tüm mimari elemanlar kendi özelliklerini bilmeli ve birbirlerine "Akıllı Eleman Mantığı" ile bağımlı olmalıdırlar. Sözgelimi bir aks taşındığında, bu aksa bağlı tüm mimari elemanlar ve ölçüleri değişmelidir.
- **Tüm mimari ve statik yapı elemanları:** Programın üç boyutlu eleman kütüphanesinde tüm mimari ve statik yapı elemanları mevcut olmalıdır. Bütün bu elemanlar kullanıcı tarafından girilen kalınlık, yükseklik, kot gibi parametrelere sahip olmalıdır.

- **Projeyi Internet üzerinden paylaşma:** Internet erişimi sağlanan her ortamda proje paylaşımı gerçekleştirilmelidir. Kullanıcılar kendilerine verilen haklar çerçevesinde ve bir proje koordinatörünün yönetiminde proje üzerinde ortaklaşa çalışabilmelidirler. Birden çok kişi hızlı ve verimli bir şekilde bir arada çalışmalıdır. Çözümlerin ortak olarak yapılabilmesi hataların en aza indirgenmesini ve zamanın daha iyi değerlendirilmesini sağlar.
- **Projeyi LAN üzerinden paylaşma:** LAN erişimi sağlanan her ofis içersinde proje paylaşımı gerçekleştirilmelidir. Internet üzerinden proje paylaşımı ile aynı özellikler sahip olmalıdır.
- **Mimari ve statik özellikleri tek bir proje üzerinde birleştirebilme:** Mimari projede ön boyutlar mimar tarafından verildiğinde, inşaat mühendisi mimarın Internet veya LAN üzerinden yolladığı proje üzerinde gerekli statik hesapları ve analizleri yapabilmelidir. Mimari ve statik özellikler tek bir proje üzerinde birleştirilmelidir. İki meslek adamı arasındaki bu eş zamanlı çalışma projenin daha verimli ve kontrollü bir şekilde ilerlemesine olanak verir.
- **Özelleştirilebilir menüler ve tuşlar:** Sık kullanılan komutlar tuşlar halinde bulunmalıdır. Kullanıcı istediği gibi menüleri ve simgeleri özelleştirebilmelidir. Mevcut araç çubuklarına komut ekleyip çıkararak veya kullanıcı tanımlı yeni araç çubukları oluşturarak komutlara daha hızlı ulaşabilmelidir. Tüm komutlara klavyeden kısa yol tuşları atanabilmelidir.
- **Tek bir program ile iki ve üç boyutlu çalışabilme, render alabilme:** Hem iki boyutlu hem üç boyutlu komutları kullanarak çalışılabilir. Çizgi gibi iki boyutlu, duvar gibi üç boyutlu elemanlardan yararlanarak kullanıcı kendi iki veya üç boyutlu elemanlarını tasarlayabilmelidir. Bu elemanlara malzeme tanımlayıp, ışık kaynaklarını düzenleyerek ve kameraları yerleştirerek render alınabilmelidir.
- **Tek bir program ile animasyon yapabilme:** Program ile animasyon hazırlanmalıdır. Bu animasyonlar AVI, JPG, BMP formatlarında kaydedilmelidir. Deneme animasyon hazırlanabilmelidir. Kaydedilen animasyonlar daha sonra film olarak seyredilmelidir.
- **Üç boyutlu mimari kütüphaneye malzeme ekleyebilme:** Üç boyutlu elemanlar için istenilen malzemeler oluşturulup tanımlanabilmelidir. Kullanıcı kendi oluşturduğu malzemenin doku, yansıma, opaklık gibi özelliklerini tanımlayıp malzeme kütüphanesine ekleyebilmelidir. Bu malzeme elemanlara atanarak istenilen özellikte render edilmiş görüntüler ve animasyonlar elde edilebilmelidir.

- **Proje kontrolü:** Proje kontrolü otomatik olarak yapılabilmelidir. Tüm elemanların kontrolü sonucunda hatalı, üst üste çakışan elemanlar sayı olarak belirlenir ve çizimde görülür. Böylece projede ortaya çıkan hatalara ilk aşamadan itibaren müdahale edilebilmelidir.
- **Ölçekli yapı elemanları:** Çizime ekleyeceğiniz yapı elemanlarını ölçekli gerçek boyutları ile görürsünüz.
- **Parametrik ölçülendirme:** Dış, iç, serbest, kesit, kot ve etiket ölçülendirme gibi tüm ölçülendirmeler otomatik olarak yapılabilmelidir. Örneğin dış ölçülendirme için, duvarı seçmek ve ölçülendirme yönünü belirlemek, duvar ile ilgili tüm ölçülerin otomatik olarak verilmesi için yeterlidir. Ayrıca ölçülendirilen elemanlarda yapılan duvar boyunun uzaması, kapı boyutunun büyümesi, pencerenin taşınması gibi değişiklikler ölçülendirmenin yenilenmesine sebep olmalıdır.
- **Metrik veya İngiliz ölçü birimi kullanabilme :** Ölçülendirme metrik (cm) veya İngiliz ölçü birimi (inch) kullanılarak yapılabilmelidir. Ölçülendirme için farklı ölçüm birimi kullanılmış olsa bile proje ortak olarak kullanılabilir.
- **Ölçü sistemi:** Program metrik veya İngiliz ölçü sistemi ile kullanılabilir olmalıdır. Ölçü birimi arayüze entegre edilmelidir. Bir birim için kullanılacak ölçü birimi arayüzden değiştirilebilmelidir. Ölçü çizgilerindeki değerler ölçü birimi ile değişmelidir.
- **Gerçek eğrisel duvar ve kapı-pencere:** Yay, çember ve elips ile gerçek eğrisel duvarlar çizilebilmelidir. Eğrisel duvarın iki ucunu farklı kotta vererek helezon tipi duvarlar oluşturulabilmelidir. Eğrisel duvarlar, kapı pencereler tam uyumlu olmalıdır. Bu tip duvarların üzerine isteğe bağlı olarak eğrisel veya düz kapı pencereler yerleştirilebilmelidir.
- **Otomatik çatı oluşturma ve detaylandırma:** Her türlü çatının oluşturulması ve detaylandırılması mümkün olmalıdır. Konturları belirlenen bir alanda çatı ve tüm elemanları otomatik olarak tanımlanabilmelidir. Tüm çatı elemanlarının genişlik, yükseklik, yerleşim düzenleri gibi özellikleri tanımlanmalıdır. Çatı üzerinde boşluklar açılabilir, boşluk açılan yerlerde çatı detaylarını oluşturulabilir ve diğer tüm detaylandırmalar mümkün olmalıdır.
- **Otomatik etiketleme :** Kapı, pencere, aks, mahal etiketleri otomatik olarak verilmelidir. Bu etiketler mimari elemanların değişimi ve eklenmesi ile otomatik olarak değişmelidirler. Örneğin döşeme alanları büyür veya küçülürse döşeme etiketleri, aksların orijine göre yeri değiştirilirse aks etiketleri değişmelidir.

- **Çok basamaklı katman kontrolü:** Tüm elemanlar kendi katmanında yer almalıdır. Kullanıcı tarafından oluşturulan çizimler için katman tanımlayabilmelidir. Daha rahat çalışmayı sağlamak amacıyla elemanlara ait alt katmanlar da olmalıdır. Bunların her birine müdahale edilebilmelidir. Elemanlara ait alt katmanlar başka katmanlarla değiştirilebilmelidir. Böylece kontrollü bir şekilde çalışma imkanı elde edilir.
- **Mimari kütüphaneye malzeme ekleyebilme:** Üç boyutlu elemanlar için istenilen malzeme oluşturulup tanımlanmalıdır. Kullanıcı kendi oluşturduğu malzemenin doku, yansıma gibi özelliklerini tanımlayıp, malzeme kütüphanesine ekleyebilmelidir. Bu malzemeyi elemanlara atayarak istediği özellikte render edilmiş görüntüleri ve animasyonları elde etmelidir.
- **Karma malzemeler oluşturabilme:** Malzemeler ve katmanlar tanımlanarak değişik malzemelerden oluşan katmanlı duvarlar, katmanlı döşemeler gibi elemanlar tanımlanabilmelidir. Bunlardan yeni tipler oluşturulabilmeli, tüm özelliklerine müdahale edilebilmeli, plan ve kesitte görülebilmelidir.
- **Gelişmiş nokta yakalama:** Tüm elemanlar düğüm noktalarına sahip olmalıdırlar. Bu noktalar elemanların birbirlerini tutmaları ve tanımları işine yarar. Bu noktalar sayesinde eleman birleşimleri hatasız olur. Kolon, giriş, duvar, çatı gibi elemanların köşeleri rahatça yakalanabilir.
- **Nokta yakalama:** Eleman üzerinde istenen mesafedeki noktayı yakalama özelliği olmalıdır. Bu sayede yakalanmak istenen nokta duvardan 20 cm içeride bile olsa yakalanabilir. Program, tıklanan nokta bir elemanın herhangi bir yakalama noktası olmadığı veya boş bir alan tıkladığında, yakalanabilecek en yakın noktadan referans uzaklığını otomatik olarak kullanıcıdan istemelidir.
- **Elemana tanımlı sağ tuş menüsü:** Bir elemana sağ tuş ile tıkladığında o elemana ait tüm komutlara ulaşılabilirdir. İsteğe bağlı olarak kullanılabilen sağ tuş ile menüye girerek bir elemana ait özellikler istenildiği gibi değiştirilebilmelidir.
- **Eleman gruplama:** Programda istenilen elemanlar seçilip "Ctrl-G" klavye kısayolu veya gruplama komutu ile gruplanabilmelidir. Böylelikle sürekli aynı elemanları seçmek için vakit kaybedilmez. İstenirse bu gruplara özel isimler de verilebilir. Gruplanan elemanlar tüm işlemlerde tek bir eleman gibi seçilir. Gerekli durumlarda grup içinde işlem yapılması istenmeyen elemanlar "Ctrl" tuşuna basılı tutulurken gruptan çıkartılabilir. İstenildiğinde her grup dağıtılabilir.
- **Eleman silme:** Program'da eleman silme çok kolay yapılabilirdir. Silmek istenilen elemanlar seçildiği zaman klavyedeki "delete" tuşuna basılması yeterli olmalıdır.

- **Seçilen elemanların gösterimi:** Program'da seçilen elemanlar, ekranda kolaylıkla fark edilmesi için, nokta nokta çizilmelidirler. Elemanlar kendi renklerini kaybetmediklerinden, seçilen elemanların nitelikleri ekranda görülebilir. Seçilen elemanların yakalama noktaları mavi renkli kutular ile belirtilmelidir..
- **Gelişmiş eleman seçim araçları:** Elemanların tümünü seçme veya seçimi tersine çevirme komutlarının olması çalışmayı hızlandıracaktır. Her elemanlar ayrı olarak seçilebileceği gibi, aynı tür elemanlar da birlikte seçilebilmelidir.
- **İstenilen elemanı bulabilme:** Çizimdeki herhangi bir eleman bulunup seçilebilmelidir. Eleman bulma komutu ile istenilen elemanı rahatlıkla ve kısa sürede bulunabilmelidir. Eğer aynı isim altında başka elemanlar da bulunuyorsa, aramaya devam edilebilmelidir.
- **Eleman yakalama:** Programın ileri eleman yakalama özellikleri olmalıdır. Uç nokta yakalama, orta nokta yakalama, kesişim noktası yakalama, dik nokta yakalama, nokta yakalama, çember merkezi yakalama, tanjant yakalama, ağırlık merkezi yakalama ve en yakın nokta yakalama gibi özelliklere sahip olmalıdır. Herhangi bir eleman yakalandığında ekranda o elemanın hangi noktasından yakalandığını belirten özel imleçler çıkmalıdır. Bu imleçlerin renkleri ve eleman yakalama işaretleri kullanıcı tarafından özelleştirilebilmelidir.
- **Eleman seçimi kaydetme:** Program seçtiğiniz elemanları unutmamalıdır. Kullanıcıya kolaylık olması için önceden seçilen elemanlar kaydedilmelidir. Gerektiği zaman önceden seçilen elemanlar tekrar çağırılabilir. Program ile daha süratli çizim yapılabilir.
- **Eleman seçimi:** Program ile eleman seçimi çok kolaydır. Eleman seçimi işlemleri için klavyeden hiçbir tuşa basmak veya komut kullanmak gerekmemelidir. Birden fazla elemanı tek tek seçmek için, fare ile elemanların üzerlerine tıklamak yeterli olmalıdır. Seçilmek istenmeyen elemanlar, üzerlerine fare ile tıklanarak kolayca bırakılabilir olmalıdır.
- **Elemanların niteliklerinin değiştirilmesi:** Seçilen elemanların nitelikleri "eleman özellikleri çubuğundan" kolaylıkla yapılabilir. Ancak daha detaylı değişiklikler için "eleman ayarları menüsünden" yapılabilir. "eleman ayarları menüsü", "eleman özellikleri çubuğunun" çalıştırılabilir. "eleman ayarları menüsü" ile seçilen elemanların değiştirilebilecek tüm özellikleri değiştirilebilir.
- **Biçim kopyalama:** Çizilen tüm elemanların katman adı, çizgi rengi, duvar kalınlığı, malzeme, genişlik, yükseklik gibi özelliklerinden istenilenleri diğer elemanlara atanabilir. Program bu özellikleri daha sonra kullanılabilmesi için, otomatik olarak kaydeder.

- **Farklı katlarda çalışabilme:** Farklı katlardaki elemanlar üzerinde aynı anda çalışabilmeli ve elemanların özellikleri aynı anda değiştirilebilmelidir.
- **Gelişmiş üç boyutlu düzenleme olanakları:** Tüm üç boyutlu elemanların özellikleri değiştirilebilmelidir. Her eleman üzerinde değişiklik yapılabileceği gibi, seçilen elemanlar üzerinde topluca değişiklik yapılabilir.
- **İstenilen ölçekte çalışabilme:** Projeler üzerinde istenilen her ölçekte çalışılabilir. Çizimler, seçilen ölçeğin ifade tekniğine uygun olarak oluşturulabilir.
- **Çizim ölçeği:** Programda kullanılan çizim ölçeği arayüze entegre edilmelidir. Ölçek menüsünden istenilen ölçek seçilebilir. Listedeki ölçekleri dışında, özel ölçekler de hazırlanabilir. Program kullanılan çizim ölçeğini tanımalıdır. Program her ölçekte kullanılan çizim tekniğini bilmeli ve kullanmalıdır. Böylece program ile farklı ölçeklerde proje çıktısı almak kolay ve hızlı bir şekilde yapılabilir.
- **Ölçek kullanımı:** Her yapı elemanı farklı ölçeklerde hangi çizim teknikleri ve kalemleri ile görüneceğini bilmelidir. Oluşturulan detaylar, farklı ölçeklerde farklı şekillerde görülmelidir. Yeni yapı elemanları tanımlanırken hangi çizim ölçeğinde nasıl görüneceği de belirlenebilir.
- **Ekran çizim ölçeği:** Program karmaşık çizimlerin ekranda daha kolay görülebileceği şekilde hazırlanmalıdır. Program için, çizimlerin ekran ölçeğine göre değişerek daha rahat çalışabilmesini sağlayan Ekran Çizim Ölçeği sistemi geliştirilmelidir. Ekran Çizim Ölçeği sistemi arayüze entegre edilmelidir. Çünkü bu sistem kullanımı kolay olduğundan tasarım sürecini hızlandırır. Örneğin, vaziyet planı ile çalışırken ekranda üst üste binmiş birçok çizgi görüleceğine, çizimin o ölçekte görülmesi gereken genel hatları görülür. Tüm kat planı ekrana sığdırıldığında, siva ve kapı/pencere detayları ekranda daha rahat görülebilmesi için, otomatik olarak ekran ölçeğine uygun çizilir.
- **Ölçekli ekran görüntüsü:** Programın ekran görüntüsü ölçekli olarak görülebilmelidir. Arayüze entegre edilmiş ekran görüntüsü ölçeği ile yapılan çizimler ekranda ölçekli olarak görülebilir.
- **AutoCAD DWG formatı ile uyumluluk :** AutoCAD DWG formatlı dosyalarını okuyup yazabilmelidir. DWG formatlı dosyaları yazarken, dosya içine sadece kendisinin okuyabileceği verileri de ekleyebilmelidir. Bu sayede DWG dosyası başka yazılımlar ile paylaşılırken kendi etiketlerini kaybetmemiş olur.
- **TTF (True Type Font) fontlarının kullanımı:** TTF fontları kullanılabilir. Paragraf şeklinde yazı yazılabilir. Yazı şekli ve büyüklüğü sonradan ayarlanabilir. Metin dosyaları program içine aktarılabilir.

- **Windows işletim sistemine uygunluk:** Program Windows işletim sistemine uygun olarak tasarlanmalıdır. Çünkü, Windows uygulamalarına alışık olan kullanıcılar programı daha kolay anlayabilir ve verimli olarak kullanabilir. Programda kullanılan komutların çalışma prensipleri popüler Windows uygulamaları ile aynı olmalıdır. Windows programlarında genel olarak kullanılan klavye ve fare kısayolları program içinde de geçerli olmalıdır.
- **Plan, kesit, görünüş ve perspektif üzerinde çalışabilme:** Plan, kesit, görünüş ve perspektif penceresi ile aynı anda çalışılabilir. Plan üzerinde yapılan değişiklikler aynı anda üçüncü boyutta görülebilmelidir.
- **İzometrik ve aksonometrik perspektif:** İzometrik ve aksonometrik perspektifler istenilen açı değerleri için hazırlanabilir.
- **Kaliteli sunum:** Program ile hazırlanan renderler, animasyonlar ve perspektifler ile kaliteli ve kolay sunumlar hazırlama olanağı vermelidir.
- **Serbest el çizimleri için dokunmatik monitör kullanımı:** Dokunmatik monitör ve dijital bir kalem yardımı ile serbest el çizimleri programa aktarılabilir. Çizim yapma ve komut alma seri bir şekilde gerçekleştirilebilir. Eskiz olarak yapılan çizimler üçüncü boyuta rahatlıkla geçirilebilir.
- **İstenilen eğri üzerinde çoklu eleman kopyalayabilme:** Her türlü eğri üzerinde, istenilen elemanlar kullanıcı tarafından tanımlanan mesafede veya sayıda çoğaltılabilir.
- **Duvar özellikleri:** Bir duvarın özelliği uzunluk, açı, kalınlık, tarama, sıva ve malzeme olarak tanımlanabilir. İstenildiğinde bu özellikler değiştirilebilir.
- **Çizimleri duvara çevirebilme:** Çizimlerle oluşturulan tasarımlar otomatik olarak duvarlara dönüştürülüp üçüncü boyuta kaldırılabilir.
- **Çizgi tipleri:** Program içinde çok sayıda çizgi tipi tanımlanmalıdır. Ama kullanıcı isterse kendi çizgi tiplerini de kolaylıkla oluşturabilir.
- **Gelişmiş üç boyutlu çizim :** Çizgi, çember, yay gibi elemanlar iki boyutlu komutlar ile hassas bir şekilde çizim yapılabilir ve düzenlenebilir.
- **Uzantıları birbiri ile kesişen iki duvarı birleştirebilme:** Uzantıları birbiri ile kesişen iki duvar, duvar uzat/böl komutu ile birleştirilebilir. Eğer duvarlar kesişemeyecek durumda ise program uyarı vermelidir.
- **Birbiri üzerinden geçen iki duvarı birleştirme:** Birbiri üzerinden geçen iki duvarı birleştirilebilir. Kesişim noktalarında birleşim pahlı olabilir. Pah çizgisel ise açı veya uzunluk değeri, eğrisel ise yarıçap değeri girilebilir.

- **Çizim açısı:** Daha kolay çizim yapabilmesi için, programın çizim açısını değiştirilebilmelidir. Çizim açısı programın arayüzüne entegre edilmelidir. İstenildiğinde açı değerine tıklanarak açılan menüden yeni bir değer seçilebilir veya klavyeden açı girilebilmelidir. Sistem açısı bir yapı elemanı referans gösterilerek de değiştirilebilmelidir.
- **Çizgi kalınlıkları:** Program ekranından gerçek çizgi kalınlıkları görülebilmelidir. Bu özellik hem çıktı ön izleme penceresine hem de programın arayüzüne eklenmelidir. Gerçek çizgi kalınlıklarını görmek, çıktıda görülemeyecek kadar yoğun detaylandırma yapmayı engelleyerek, teknik çizim süresini kısaltır. Paftanın yoğunluğuna göre çizimler düzenlenebilir.
- **Çizgi tipi oluşturma:** Program ile istenilen çizgi tipleri oluşturulabilmelidir. Çizilen herhangi bir çizgi tip veya desen, çizgi tipi olarak belirlenebilmelidir. Oluşturulan yeni çizgi tipleri diğer projelerde de kullanılabilir.
- **Mimari çizim teknikleri:** Program mimari çizim tekniklerini kullanmalıdır. Duvar birleşimleri otomatik olarak yapılmalıdır. Farklı malzemelerden yapılmış veya yükseklikleri eşit olmayan duvarlar birleşmemelidir. Aynı malzemedeki yapılmış duvarlar kesiştiğinde, düğüm noktaları otomatik olarak temizlenir olmalıdır.
- **Duvar birleştirme:** program ile iki duvar üç farklı şekilde uzatılarak birleştirilebilir. Her iki duvarın boyları kesişim için yeterli değil ise uzar. Eğer biri kesişimden daha uzunsa, boyu kısılır. Eğer birisinin uzaması istenmiyor ise, sadece diğeri uzatılabilir.
- **Otomatik mahal belirleme:** Duvar ve giriş ile kapatılmış alanlarda döşeme otomatik olarak oluşturulabilmelidir. Döşeme bilgileri otomatik olarak verilebilmelidir. Mahaller için malzeme, mahal isimleri, sıva alanlarının toplam alanı etkileyip etkilememesi gibi bilgiler kullanıcı tarafından belirlenebilmelidir.
- **Otomatik numaralandırma:** Aks, kolon, giriş, temel gibi tüm elemanlar otomatik olarak numaralandırılmalıdır.
- **Projenin kolaylıkla antetli kağıda yerleştirilebilmesi:** İstenilen proje düzeni oluşturulup, çizimler bu düzene göre antetli kağıda yerleştirilebilmelidir. Belirlenen proje düzeni dosyaya kopyalanarak istenilen projede çağırılıp kullanılabilir.
- **Siyah-beyaz veya renkli baskı ön izleme :** Windows işletim sisteminin baskı alma özelliği kullanılmalıdır. Her türlü yazıcı ve çizici ile uyum sağlanmalıdır. Baskı ön izleme ile, yazılanlar kontrol edilebilmeli, kalem kalınlıkları ayarlanabilmelidir.

- **Kat yüksekliđi ile kot iliřkisinin otomatik olarak ayarlanması:** Kat yüksekliđi ve kot iliřkisini program otomatik olarak ayarlayabilmelidir. Yükseklik deđiřtirildiđinde kot deđerı, kot deđiřtirildiđinde yükseklik deđerı otomatik olarak dzenlenmelidir. Ařađı ve yukarı kat ekleme komutu ile yeni katlar ekleme kolay olmalıdır.
- **Son yapılan iřlem:** Son yapılan iřlem geri alınabilmelidir. Geri alma iřlemi dosyanın ilk ađılmıř haline kadar tekrarlanabilmelidir. Bu iřlemler bakıř ađılarındaki deđiřiklikleri içermemelidir. Son yapılan iřlem bir komut ile tekrar edilebilmez.
- **Proje üzerinde not alma:** Proje üzerine notlar almak, iřaretler koymak gibi çizime ait olmayan, fakat ilerde çizimle birlikte kullanılabilcek çalıřmaların yapılabildiđi özel bir katman olmalıdır.
- **Herhangi bir iki boyutlu elemanı üç boyutlu hale dönüřtürebilme:** Çizilen bir iki boyutlu eleman üç boyutluya dönüřtürölerek mimari kütüphaneye eklenebilmelidir. Özellikleri istenilen oranlarda deđiřtirilerek yeni elemanlar yaratılabilmeli ve saklanabilmelidir.
- **Hakediř hesapları:** Tüm hakediř hesapları program tarafından otomatik olarak yapılmalıdır.
- **Hakediř hesapları için istenilen řekilde detaylı tablolar oluřturabilme ve Excel'e veya bir veritabanına veri aktarılabilme:** Kullanılan malzemeler, malzeme fiyatları, çalıřılan iř günü sayısı gibi bilgileri hakediř hesaplarında kullanmak için istenilen řekillerde detaylı tablolar oluřturulabilmelidir. Bilgilerin dzenlenmesinde esneklik sađlamak amacıyla, tabloların içine konacak bilgilerin boyutuna göre basit tablolar veya çok sayıda bilgi alabilecek matris tablolar yapılabilmelidir. Hakediř tabloları, bir veri tabanına veya Excel programına veri aktarabilmelidir.
- **Hakediř hesaplarının otomatik olarak güncellenmesi :** Projede herhangi bir eklenti yapıldıđında veya herhangi bir eleman projeden çıkartıldıđında hakediř hesapları otomatik olarak güncellenmelidir. Örneđin bir kapı projeden silindiđinde veya deđiřtirildiđinde, bu kapının özellikleri hakediř hesaplarında otomatik olarak güncellenmelidir. Bu sayede bu hesaplar ile ilgili olarak yapılması gereken rutin düzeltmeler ve bunlara harcanacak vakit ve emek ortadan kaldırılmıř olur.
- **Herhangi bir eleman üzerine veri tablosu ekleyebilme :** İstenilen bir elemanın üzerine onun özelliklerini anlatan veri tabloları bırakılabilmelidir. Eleman üzerine gidilip sađ tuřa basılı tutulduđunda bu tablolar görünür hale gelmelidir.

- **Özelleştirilebilir eleman etiketleri oluşturabilme:** Elemanlara bağlı olan, onların özelliklerini içeren veri etiketler otomatik olarak oluşturulmalıdır. Elemanların özellikleri değiştirildiği zaman bu etiketler de otomatik olarak güncellenmelidir.
- **Elektronik eğitim merkezi ile Internet'ten eğitim desteği :** Bir yardım merkezinden 24 saat destek alınabilmelidir. Özellikle programı yeni öğrenmeye başlayan kullanıcıların akıllarına takılan sorulara çözüm getirmek amacıyla bir Internet sitesi kurulmalıdır.
- **Program ile beraber verilen eğitim CD'si ile eğitim olanağı:** Program ile birlikte gelen eğitim CD'si ile kullanıcılar çizim tekniklerini ve komutlarını öğrenebilmelidirler. Çizimlerin nasıl daha verimli yapılacağı hakkında bilgi edinebilmelidirler. Kullanıcılar programın özelliklerini kendi başlarına öğrenebilmelidirler.
- **Ücretli CD'ler ile ileri eğitim olanağı:** Özel konularda ilerlemek isteyen kullanıcılar için, farklı konularda hazırlanmış CD'ler hazırlanmalıdır.
- **Tartışma grubu:** Kullanıcılar bu gruba Internet üzerinden katılarak, başta program hakkındaki görüşleri olmak üzere her türlü konuyu gündeme getirebilmelidirler.
- **Akıllı yapı elemanları:** Programda kullanılan tüm yapı elemanları, bilgisayar tarafından tanınan akıllı eleman olmalıdırlar. Program ile beraber gelen standart yapı elemanları projeye uygun olarak özelleştirebilmelidirler.
- **Özelleştirilmiş duvar tipleri:** Kullanıcı tanımlı duvar tipleri belirlenebilmelidir. Gerçek yapı malzemeleri ile simüle edilmiş gerçekçi duvarlar ile projeler hazırlanabilmelidir. Bu uygulama aşamasında en az hatalı üretimi en kısa sürede belirlenen maliyetler çerçevesinde yapılmasına olanak kılar.
- **Özelleştirilmiş mimari elemanlar:** Program ile özel mimari elemanlar kolaylıkla üretebilmelidir. Projeye uygun özel mimari elemanların kısa sürede üretebilmesi, tasarımların bilgisayar ortamında daha kolay modellenmesini sağlar. Bu şekilde mimari detaylar üç boyutlu olarak daha kolay hazırlanabilir, hakediş dosyaları daha gerçekçi olur. Üretimdeki hatalar en aza iner, zamandan ve paradan tasarruf edilir.
- **Animasyon hazırlama:** Kısa bir sürede projenin animasyonları hazırlanabilmelidir. Program içindeki hazır sunum şablonları kullanılarak proje sunumu hazırlanabilmelidir. Kullanıcı tanımlı sunum şablonları da hazırlanabilmelidir.
- **Otomatik Detaylandırma:** Proje otomatik olarak detaylandırılmalıdır. Program içinde kullanılan akıllı yapı elemanı teknolojisi sayesinde her yapı elemanı birbirlerine nasıl davranacağını bilmelidir. Yapı elemanlarının birleşim noktalarında olası detaylar otomatik olarak üretilebilmelidir.

- **Program kullanımı:** Program kullanımı çok kolay ve basit olmalıdır. İhtiyaç duyduğu anda kullanması için hazırlanan el kitapları, her seviyedeki kullanıcının kolaylıkla anlayabileceği bir dilde yazılmalıdır. Konuşma diline yakın bir şekilde yazılmalı, anlaşılabilir olması açısından kısa cümleler kullanılmalıdır. Yardım alınan konularda olaylar nedenleri ile anlatılmalıdır.
- **Program Açılışı:** Program çalıştırıldığında en son çalışılan projeyi açar. Büyük projelerin aktif olan son paftası yüklenerek, vakit kazandırır.
- **Yardım kitapları:** Programın iki ayrı yardım kitabı olmalıdır. Birinci yardım kitabı programda kullanılan komutları açıklamalıdır. İkinci yardım kitabı ise program ile mimari proje üretim sürecinde kullanılacak yollardan bahsetmelidir.
- **Kullanım yöntemleri:** Yardım kitaplarında programın kullanım yöntemleri detaylı olarak açıklanmalıdır. Kullanıcıya anlatılan tüm komutlar, bilgisayar destekli mimari tasarım sürecine uygun olarak hazırlanmalıdır.
- **Yardım kitaplarının tasarımı:** Yardım kitaplarının sayfa tasarımı, yazı karakterleri ve büyüklükleri, kullanılan anlatım dili ve mimari çizimler, kullanıcıların anlatılanları daha iyi öğrenebilmeleri için özel olarak hazırlanmalıdır.
- **Kullanıcı arayüzü:** Program Windows kullanıcılarının alışık olduğu bir arayüze sahip olmalıdır. Tüm komutlar ve menüler özelleştirilebilmelidir.
- **Arayüz renkleri:** Program arayüzünde farklı renkler kullanılmalıdır. Uzun süre bilgisayar başında kalan kullanıcılar için, gözü yormayan farklı arayüz rengi paletleri olmalıdır. Her program renk paleti, farklı arka plan renkleri için geliştirilmiş uygun çizgi renkleri kullanılmalıdır.
- **Arayüz maskeleri:** Program ilk kez kurtulduğunda arka planda beyaz rengin olduğu maske yüklenir. Program farklı maskeleri olmalıdır. Kullanıcılar kendi renk paletlerini oluşturabilmeli ve diğer kullanıcılar ile paylaşabilmelidirler. Kullanıcıların oluşturduğu diğer renk paletleri program web sitesinden alınabilmelidir.
- **Kullanılan komutlar:** Programda kullanılan komutlarda anlam kargaşası olmamalıdır. Komutlar kolaylıkla anlaşılabilir olmalıdır.
- **Ergonomik kullanım:** Kullanıcıların daha ergonomik çalışabilmesi için, sık kullanılan komutlar belirlenerek, en kısa sürede erişilecek şekilde yerleştirilmelidirler. Komut ikonları özel olarak tasarlanmalıdır. Komutların ne oldukları, ikonlarından kolaylıkla anlaşılabilir olmalıdır.

- **Hatalı kullanım:** Herhangi bir komut kullanılmaya başlandığında, program kullanıcın hareketlerini izlemelidir. Program için özel olarak geliştirilmiş bir sistem, hatalı kullanımlarda kullanıcıyı bilgilendirerek yapılmak istenenleri doğru bir şekilde ve en kısa sürede yapılmasını sağlamalıdır.
- **Parametre girişi:** Program yanlış veya eksik parametre girildiğinde kullanıcının hatasını otomatik olarak tespit etmelidir. Kullanıcının yapmayı düşündüğü ama yapamadıklarını otomatik olarak anlamalı ve bunu kullanıcıya bildirmelidir.
- **Parametre kullanımı:** Parametre girişi kolay olmalıdır. İstenilen uzunlukta ve açıda çizgi veya duvarlar kolaylıkla çizilebilmelidir.
- **Parametrik program:** Program parametrik olmalıdır. Program ilk kez çalıştırıldığında, “günün ipucu” (tip of the day) adında açılan bir pencerede parametrik çalışma anlatılmalıdır.
- **Program logosu:** Program çalıştırıldığında program logosu ekrana gelmelidir. Program arka planda yüklenip kullanıma hazır hale gelmeden, program logosu ekrandan kalkmamalıdır. Program logosu her versiyon için özel olarak hazırlanmalıdır. Logo içinde program ile yapılmış mimari projelerden örnekler olmalıdır.
- **Sıva kullanımı:** Program projenin her aşamasında sıvalı veya sıvasız olarak gösterebilmelidir. Proje çizimleri sıvasız olarak hazırlanıp, uygulama çizimleri sıvalı olarak devam edilebilmelidir.
- **Seçim araçları:** Programın gelişmiş eleman seçim araçları olmalıdır. Karmaşık çizimlerde istenilen elemanlar kolaylıkla seçilebilmelidir. Pencere açarak elemanları seçerken, sadece pencere içine giren elemanlar seçilebileceği gibi, pencere içinde kalan tüm elemanlar da seçilebilir. Seçim hattı belirleyerek, bu hat üzerinde kalan elemanlar seçilebilmelidir.
- **Filtreleme:** Seçilen elemanların filtrelenmesi çok kolaydır. Filtreleme arayüze entegre edilmelidir. Filtrelenecek karmaşık çizimlerde istenilen elemanlar filtrelenerek seçilebilir .
- **Komut Kullanımı:** Programın komut kullanımı esnek olmalıdır. Kullanıcı komutları iki türlü kullanabilmelidir. Popüler Windows uygulamaları gibi önce eleman seçilip sonra komut kullanabilir veya eleman seçimi komut alındıktan sonra da yapılabilir.

- **Objeye duyarlılık komutu:** Program OSC (Objeye duyarlılık komutu) metodu ile projelerde daha kolay ve hızlı deęişiklik olanaęı sağlayabilmelidir. OSC metodu, çizilen elemanların komut kullanılmadan taşınabilmesini, kopyalanabilmesini, çekiştirilebilmesini, uzatılabilmesini ve niteliklerinin deęiştirilebilmesini sağlar. Seçilen elemanı taşımak için shift tuşuna basarak sürükleyip bırakmak yeterli olacaktır. Sürükleyip bırakırken Ctrl tuşuna basılırsa seçili elemanlar kopyalanır. program'da seçilen elemanların yakalama noktaları ekranda aktif olur. Yakalama noktalarını sürükleyerek elemanlar çekiştirilip, boyları deęiştirilebilir.
- **Kesme komutu:** Programın kesme (trim) komutu ile her türlü eleman kesilebilmelidir. Plan düzleminde duvarların uzunlukları kesilebileceęi gibi, perspektif penceresinde duvarların yükseklikleri de kesilebilmelidir..
- **Kapat komutu:** Programın "kapat" komutu, üç tarafı çizilen mekanları kapalı hacimlere dönüştürebilmelidir. "kapat" komutu duvar, giriş, çatı, döşeme ve çizgiler ile çizilmeye başlanmış hacimleri kapatabilmelidir.
- **Aks takibi komutu:** Program ile daha hızlı yaylar oluşturulması için, "aksı takip et" komutu tasarlanmalıdır. Böylece çizim araçları ile çizim yaparken sağ tuşa basıldığında açılan menüdeki "aksı takip et" komutu ile seçilen aks üzerine yapışarak giden çizim yapılabilir. Seçilen aks çizgi, duvar kenarı, çember veya ızgara çizgileri gibi ekrandaki her türlü çizim olabilir.
- **Düzenleme komutları:** Programın düzen menüsü içinde tüm Windows uygulamalarında geçerli olan kes, kopyala ve yapıştır komutları bulunmalıdır. Kes, kopyala ve yapıştır komutları klavyeden standart Windows kısayol tuşları ile kullanılabilir. Bu komutlar aynı proje içerisinde farklı katlar arasında kullanılabildeęi gibi, dięer projeler arasında da kullanılabilir.
- **Komut grupları:** Programın komutları işlevlerine göre gruplandırılmalıdır. Yapı elemanları oluşturma komutları ve düzen komutları ayrı komut grupları olarak ayrılmış olmalıdır. Her bir komut grubu sık kullanılanlar ve az kullanılan olarak ikiye ayrılmalıdır. Program yeni yüklendiğinde sık kullanılan komut menüleri ekrana gelmelidir. Kullanıcı kendi komut grubunu oluşturabilmelidir. Ekranda sağ tuşa basıldığında açılan menüde, o bölgede kullanılabilecek en sık komutlar bulunmalıdır.
- **Pahlama komutları:** Köşeli ve yuvarlak pahlama komutları ile hem çizgiler hem de yapı elemanları pahlanamalıdır. Duvar birleşimleri veya döşeme köşeleri de pahlanamalıdır

- **Döndürme komutu:** Programın döndürme komutu iki farklı şekilde kullanılabilir. Döndürülecek elemanlar seçilip döndürme eksenini noktası belirlendiği zaman koordinat kutusundaki açı değeri aktif olur. Klavyeden girilecek ilk sayı değeri açı değeri olur. Eğer açı değeri girilmezse referans vererek döndürme yapılabilir.
- **Kullanılan komutlar:** Programda kullanılan komutlar iptal edilmediği sürece aktif olmalıdırlar.
- **Yakınlaştırma, uzaklaştırma komutları:** Programın yakınlaştırma, uzaklaştırma ve kaydırma komutlarının klavye kısa yolları oluşturulmalıdır. Klavyedeki kestirme tuşlar ile çizim ekranında istenilen yerlere kolaylıkla erişilebilmelidir. Klavyedeki "+" ve "-" tuşları ile zoom yapılır. Ara çubuğa sürekli basıldığında ekranı kaydırma komutu aktif olmalıdır. Program çizim ekranındaki kaydırma çubukları ile ekranı kaydırılabilir. Programın yakınlaştırma komutu klavyeden "z" harfi ile çalışır.
- **Yakınlaştırma komutu:** Ekranda seçilen alan yakınlaştırma komutu ile büyütüldüğünde, seçilen alan ekrandan en az %5 daha küçük olmalıdır.
- **Ekrana sığdır komutu:** Programın ekrana sığdır komutu ile projede çizilen tüm elemanlar ekran görüntüsü içine sığdırılmalıdır. Ekranın köşelerindeki elemanların daha kolay seçilebilmesi için, ekranın köşelerinde pay bırakılmalıdır.
- **Renk Paleti:** Programın renk paleti özel olarak hazırlanmalıdır. Kullanıcının gözünü yormayan, ekranda kolay seçilebilen renkler kullanılmalıdır. Duvar, kolon, taramalar, tefriş elemanları ve yazılar gibi tüm mimari elemanlar için özel renkler belirlenmelidir. İstenildiğinde program renk paleti dışında Windows'un 16 milyon renkli paleti de kullanabilmelidir.
- **Obje özellikleri çubuğu:** Programın arayüzüne entegre edilmiş "obje özellikleri menüsü", seçilen elemanların özelliklerini göstermelidir. Renk, çizgi kalınlığı, katman, tarama şekli gibi özellikler çekme menülerden hızlı bir şekilde değiştirilebilmelidir.
- **ESC tuşu:** Program ile seçilen elemanlar "esc" tuşu ile bırakılmalıdır. Program içinde açılan tüm menüler ve alınan tüm komutlar "esc" tuşu ile iptal edilmelidir.
- **Yardım menüsü:** Program'da komutların doğru kullanılabilmesi için arayüze entegre edilmiş yardım menüsü bulunmalıdır.
- **Tarama Tipleri:** Programın tarama tipleri, dünya ve Türkiye standartlarına uygun olarak hazırlanmalıdır. Duvar, kolon, döşeme gibi yapı elemanları oluşturulduğunda taramalar otomatik olarak çizilmelidir. Çizilen tarama tiplerinin yoğunluğu çizim ölçeğine göre değişmelidir.

- **Tarama tipi oluřturma:** Program ile istenilen tarama tipleri kolaylıkla oluřturulabilmelidir. Çizilen herhangi bir desen, "tarama tipi oluřtur" komutu ile programın tarama tiplerine eklenebilmelidir. Oluřturulan tarama tipleri diđer projelerde de kullanılabilir.
- **Resim dosyaları ile tarama:** Program ile gelen en yeni özelliklerden birisi, resim dosyalarının da tarama tipi olarak kullanılabilmesi olmalıdır. Sunuřlar için ıslak hacimlere seramik veya yařama alanlarına parke resimli tarama yapılabilir. Resimli taramalar istenildiğinde çizgili taramalara dönüřtürülmelidir.
- **Çoklu proje açma özelliđi:** Program aynı anda birden fazla proje açmalıdır. Her yeni proje ayrı bir program penceresinde açılarak, kullanıcının projeleri karıřtirmasını engellemelidir. Projeler arasında çizilen elemanların transferi yapılabilir.
- **Mimari düzenlemeler:** Program'da aynı tür düzenleme (edit) işlemleri için genel komutlar kullanılmalıdır. Tüm düzenleme komutları her yapı elemanı için geçerli olmalıdır. Uzatma (extend) veya kesme (trim) gibi komutlar duvar, kiriř, döřeme gibi tüm çizim elemanları ile kullanılabilir olmalıdır. Düzenleme komutları, plan düzlemleri ve kesit / görünüşler ile beraber tüm perspektif açılarında kullanılabilir. Program'daki tüm düzenleme (edit) ve çizim komutları kesit / görünüş ve perspektif açılarında kullanılabilir olmalıdır.
- **Kısa yollar:** Program'da klavye kısa yolları, kullanıcıların daha hızlı çalışabilmeleri için özel olarak hazırlanmalıdır. Klavyedeki tuř kombinasyonları farklı amaçlara yönelik olmalıdır. Klavyedeki ara çubuk, duvar çizilirken çıkıř hattını deđiřtirmek için kullanılmalıdır.
- **Izgara aralıkları:** Çizim kolaylıđı için programın ızgara aralıkları kullanılabilir. Izgara aralıkları menüsü kolay kullanılabilmesi için arayüze entegre edilmiř olmalıdır. Izgara komutunun aktif olup olmadığı kolayca anlaşılabilir. Izgara komutu tıklanarak açılıp kapatılabilir. Sađ tuřa ile tıklanıđında açılan menüden ızgara ayarlarına girilir. Izgara ayarlarına programın araçlar araç çubuğundan girilebilir. Izgara ayarları menüsünden x, y, z koordinatlarında ızgaraların öteleneceđi mesafeler girilmelidir. Izgaralar sistem açısına otomatik kilitlenir olmalıdır. Izgara açısı, istenildiğinde ızgara menüsünden deđiřtirilebilir.

- **Izgara kullanımı:** Mimari çizimlere uygun farklı tiplerde ızgaralar kullanılabilir. Çizilen çizgiler veya yaylar, ızgara menüsündeki "çizimden ızgara aksı al" komutu ile ızgaraya dönüştürülebilir. Böylece, ızgaradaki çizgiler düzlemde sonsuza kadar giden vektörler haline gelir. Kırıklı çizgiler veya daire parçaları, oldukları boyutlarda ızgara haline gelirler. Izgara çizgileri tüm düzenleme komutları ile normal çizim elemanları gibi kopyalanabilir, taşınabilmeli, silinebilmeli veya kesilebilmelidir. Izgara elemanlarının düzenlenebilmesi için, ızgara düzenleme komutu kullanılmalıdır. Izgara düzenleme komutu, en kısa yoldan ulaşılabilmesi için, sağ tuş menüsüne konulmalıdır.
- **Izgara yakalama:** Programın çizim araçları ile çizilen her eleman ızgaraları yakalanabilir. Bunun için eleman yakalama komutu aktif olmalıdır. Eleman yakalama komutu ile ızgaraların kesişim noktaları, uç noktaları ve çember merkezi noktaları ile ızgara üzerinden herhangi bir nokta yakalanabilir.
- **Ekran yenilenmesi:** Programın çizim ekranı sürekli tazelenmelidir. Diğer CADD programların aksine, silinen veya değiştirilen elemanların ekranda izi kalmayacak şekilde çalıştırılabilir.
- **Kopyalayarak aynalama:** Seçilen elemanlar kopyalanarak aynalanabilir.
- **Çoklu kopyalama:** programın çoklu kopyalama komutu, istenen aralıklar ile aynı elemanların farklı yerlere hızlı olarak kopyalanmasını sağlar. Düzen menüsü altındaki çoklu kopyalama komutu, klavyeden Ctrl+c+shift kısa yolu ile kullanılabilir. Klavyedeki esc tuşu çoklu kopyalama komutu kullanımını bitirir.
- **Kopyalayarak kaydırma:** Düzen menüsündeki kopyalayarak kaydır komutu, seçilen elemanları istenen aralıklarda ve sayıda kopyalamayı sağlamalıdır. Kopyalayarak kaydır komutu ile elemanlar ızgara üzerinde veya çember üzerinde kopyalanabilir. Kopyalayarak kaydır komutu ile açılan menü iki sayfadan (tab) oluşur. Birinci kısımda x,y,z eksenlerinde ızgara üzerinde kopyalama yapılır. Izgara kısmına x,y,z eksenlerinde kopyalama mesafeleri ve kopyalama sayıları girilir. Izgaranın düzlem ile yapacağı açı değeri değiştirilebilir. İkinci kısımda dairesel kopyalama yapılır. Yarıçapı verilen çember aksı üzerinde elemanlar istenilen sayıda kopyalanabilir, istenilen aralıklarla veya açılar ile dizilebilir. Elemanlar çember üzerinde görüldüğü gibi kopyalanabilirler veya merkeze doğru çevrilebilirler.
- **Çoğaltma:** Programın çoğalt (clone) komutu ile seçilen elemanların üzerlerine kopyaları çıkartılabilir. Üst üste oluşan iki eleman ayrı ayrı kullanılabilir.

- **İstenilen mimari elemanları başka katlara kopyalayabilme:** Kat bazında kopyalama yaparken, seçilen mimari elemanların yanı sıra tüm elemanlar tek bir komut ile kopyalanmalıdır. Seçilen tüm elemanları tüm katlara aynı anda kopyalamak gereklidir.
- **Dosya temizliği:** Program çalışılan projeyi sürekli olarak kontrol etmelidir. Çoğaltma komutu veya başka komutlar ile üst üste gelen ve birbirinin aynı olan elemanlar, tek bir eleman olarak gösterilebilmelidir. Böylece hakediş dosyalarında fazla ölçüm çıkmaz. Ayrıca dosya menüsünde bulunan dosya ayarları açılır menüsüne “dosya temizle” komutu konulmalıdır. Bu komut ile üst üste gelen elemanlar temizlenebilmelidir.
- **Offset komutları:** Programda iki ayrı offset komutu olmalıdır. Düzen menüsü altındaki offset komutu ile ekranda çizilmiş her türlü eleman, çizgi veya çember olarak offsetlenmelidir. “elemanları Offsetle” komutu, mimari eleman çizimi komutları içinde olmalıdır. Bu komutla tüm mimari elemanlar offsetlenebilir. Örneğin duvarlar veya döşemeler, kirişler ile offsetlenebilir.
- **Offset kullanımı:** Programın offset komutu kullanıcının kullanımına göre iki ayrı şekilde çalışmalıdır. Kullanıcı offset komutunu aldıktan sonra eleman seçmeden mesafe verebilmelidir. Böylece elemanlar seçildiği zaman offsetlenir. Eğer offset mesafesi verilmeden elemanlar seçilirse, mesafe girildiği elemanlar offsetlenebilir.
- **Program Dili:** Program, aralarında Türkçe de olan sekiz ayrı dünya dilinde yazılmış olmalıdır. Tüm komutlar ve yardım dosyaları kullanılan dillere göre özel olarak tercüme edilmelidir. Bilgisayar terimlerinin, tercüme edilebileceği dillerde kullanımı araştırılmış olmalıdır.
- **Deneme programları:** Programın deneme programları özel olarak hazırlanmalıdır. Programı test eden ilk kullanıcının bütün hareketleri “program öğreticisi” tarafından izlenmelidir. Test aşamasında kullanıcıya kullandığı komutlar ile ilgili tavsiyeler vermelidir. Verilen tavsiyeler ekranda açılan yardım penceresi şeklinde olmamalıdır. Ekranda komutların kullanımı Öğretici tarafından uygulamalı olarak gösterilmelidir. Öğretici komutları sesli olarak anlatırken, ekrandaki küçük bir pencerede yazılı bilgi verilmelidir.
- **Yeni yapı elemanı tanımlama:** Yeni bir yapı elemanı sadece profil kesiti çizimi ile tanımlanabilmelidir.

- **Koordinat kutusu:** Program'da parametre girişleri koordinat kutusundan yapılmalıdır. Koordinat kutusu ekranda yüzen bir menü şeklinde olmalıdır. Kullanıcı isterse koordinat kutusunu arayüze entegre edilebilmelidir. Koordinat kutusu ölçekli çizim yapmayı kolaylaştırır. İstenilen uzunluk, yükseklik, genişlik, açı veya x,y,z koordinatları koordinat kutusuna girilir. Parametre girişlerinin daha hızlı olabilmesi için klavye kısa yolları vardır. "L" uzunluk, "A" açı, "W" genişlik, "H" yükseklik, "X" x koordinatı, "Y" y koordinatı için kullanılır. Program ile çizim yaparken uzunluk değeri koordinat kutusunda aktif olur. Uzunluk değerini girmek için klavye kısa yol tuşu kullanılması gerekmez. Klavyeden girilen ilk sayı değerini program uzunluk değeri olarak algılar.
- **Koordinat kutusu açı değeri:** Koordinat kutusundaki açı değeri istenilen açıda çizim yapmayı sağlamalıdır. Açı değeri verildiği zaman ekrandaki kursör verilen açı değerine kilitlenmelidir.
- **Program ile ilgili gelişmeler:** Program ile ilgili haftalık gelişmeler "bugünkü BDT haberleri" adlı bir Internet sitesinden takip edilebilmelidir. Ayrıca programın kayıtlı kullanıcılarına yazılı olarak postalanmalıdır. Bu magazin içinde program tavsiyeleri ve gelişmeleri anlatılmalıdır.
- **Komut kullanımı:** Programda herhangi bir komut kullanılmıyorsa kursör eleman seçme konumunda olmalıdır.
- **Proje dosyası:** Programda hazırlanan her proje tek bir dosya adı ile kaydedilmelidir. Her proje için tek bir dosya kullanımı proje yönetimini kolaylaştırır. Dosya yedeklemesi daha kolay olur. Internet veya LAN üzerinden dosya paylaşımı daha rahat olur.
- **Internet tarayıcı:** Program içinde Internet tarayıcı programı olmalıdır. Programın "web tarayıcısı", kullanıcıların program web sitesine bağlanmasını sağlamalıdır. Bu program ile, program güncellemeleri, arayüz maskeleri gibi yamaları otomatik olarak alınabilir. Ayrıca diğer program kullanıcıları ile konuşma odalarında tartışılabilir veya program haber grubuna mesaj bırakılabilir.
- **Rampa modelleme:** Düz veya döner rampa modellenenmelidir. Rampalara istenilen kalınlık ve malzeme atanabilmelidir.
- **Program logosu:** Masa üzerindeki programın logosu her sürüm için farklı olmalıdır . Her farklı logo ayrı bir sürümü simgelemelidir.

5.2 İş Özeti

Bu kavramsal mimari yazılım arayüzü modeline i-CAD adı verilmiştir. Arayüz modeli için Delphi IDE (Integrated Development Environment) kullanılmıştır (Williams, Walmsley, 1998). Delphi IDE, öğrenimi kolay olan ve konuşma diline benzerliği ile tanınan Pascal programlama dilini kullanmaktadır (Türkoğlu, 2001).

Arayüzü çalıştıran çizim motoru için PowerCAD adlı Delphi eklentisi kullanılmıştır. PowerCAD eklentisi ile gelen tüm çizim elemanları Delphi üzerinde açılan i-CAD formuna araç çubukları olarak konulmuştur. Çizim yapılacak alan için PowerCAD eklentisinin TPowerCAD eklentisi kullanılmıştır (45).

i-CAD içinde açılan menüler, Delphi ile gelen standart Windows menü eklentileri ile yapılmıştır. i-CAD için kullanılacak "CAD" adlı dosyalar Windows menülerine eklenmiştir. i-CAD içinde açılan yüzer özellikler araç çubuğu için TPowerCAD eklentisi kullanılmıştır. Çekme menü komutları, i-CAD içinde kullanılan ve kullanılacak komutlar için düzenlenmiştir.

Delphi ile oluşturulan çalışır i-CAD programının "Kur" dosyası InstallShield Express adlı bir program ile oluşturulmuştur (22). i-CAD'in yardım dosyaları Help&Manual adlı program ile hazırlanmıştır (18). Programın kısıtlı süreli sürümü yardım dosyalarına reklam metinleri eklemektedir. Help&Manual programının hazırladığı hpj ve cnt dosyaları Microsoft firmasının ücretsiz olarak dağıttığı Microsoft Help Workstation programı ile derlenmiş, hlp ve cnt dosyaları oluşturulmuştur (31).

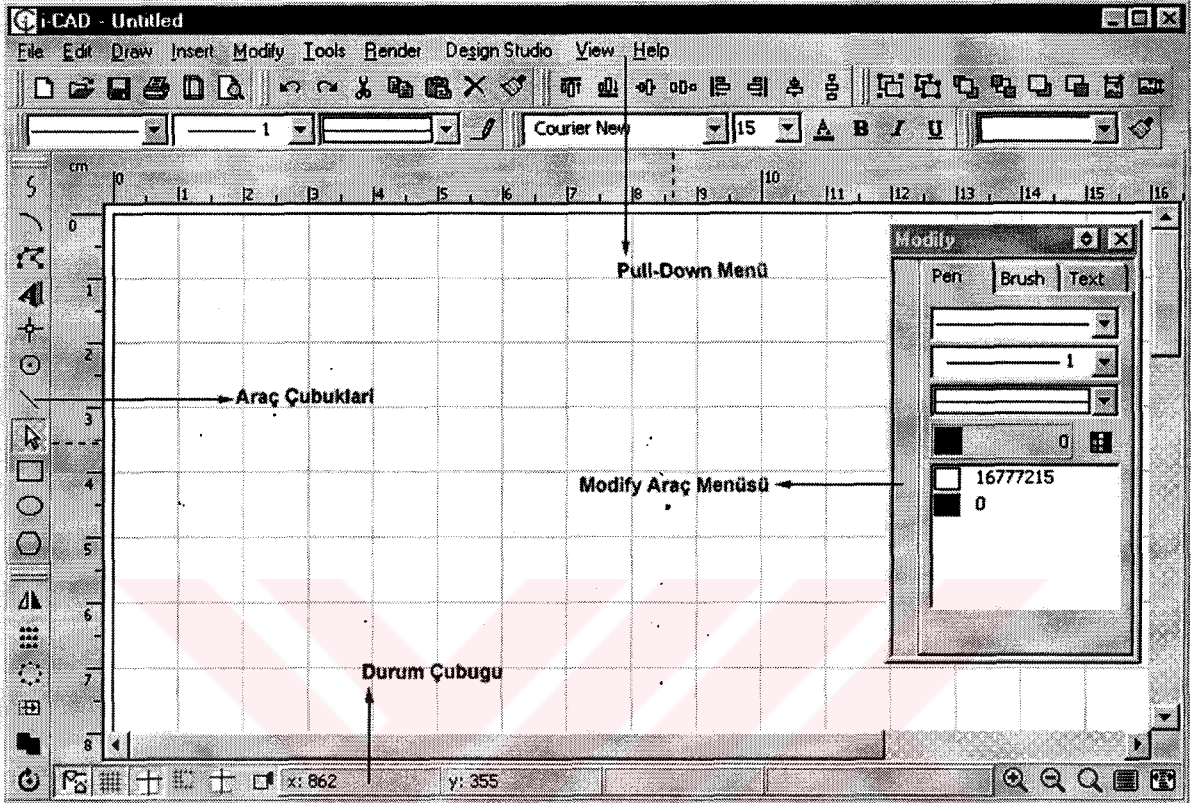
Bilgisayar destekli yazılımlarda kullanılan komutların Türkçe karşılıkları henüz belirlenmemiştir. Yabancı dildeki bilgisayar terimlerini Türkçe'ye çevirmekte olan Türk Bilişim Derneği de, BDT yazılımlarında kullanılan komutları henüz Türkçe'ye çevirememiştir (51). Türkçe olarak satılmakta olan her BDT yazılımı, kendi Türkçe karşılıklarını kullanmaktadır. Bu durum komut isimlerinde çelişki yaratmaktadır. Bu yüzden i-CAD yazılımının dili İngilizce'dir.

5.3 Kullanıcı Arayüz Önerisi

Kullanıcı arayüzü dört bölgeye ayrılmıştır (Şekil 5.1):

- 1) Çekme menüler (Pull-Down Menü)
- 2) Araç çubukları

- 3) Durum çubuğu
- 4) Modify Araç menüsü



Şekil 5.1 i-CAD ekranı

5.3.1 Çekme Menüler

Çekme menüleri, şu komutlardan oluşmaktadır:

- 1) File (Dosya açma)
- 2) Edit (Düzenleme)
- 3) Draw (Çizim)
- 4) Insert (Dosya ekleme)
- 5) Modify (Değiştirme)
- 6) Tools (Araçlar)
- 7) Render

- 8) Design Studio (Tasarım stüdyosu)
- 9) View (Görünüm)
- 10) Help (Yardım)

File, edit, view ve help menüleri, yazılımlarda kullanılan standart başlıklardır. Draw, insert, Modify, Tools, Render ve Design Studio başlıkları ihtiyaç listesine göre hazırlanmıştır (Şekil 5.2).



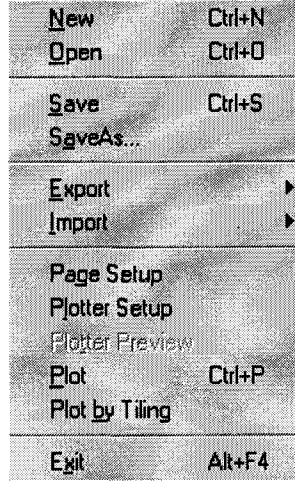
Şekil 5.2 Çekme menüler

5.3.1.1 File

File menüsü şu komutlardan oluşmaktadır (Şekil 5.3):

- 1) New: Yeni dosya açar.
- 2) Open: Dosya açar.
- 3) Save: Dosya kaydeder.
- 4) SaveAs: Dosyayı farklı isimde kaydeder.
- 5) Export: Farklı formatlarda dosya kaydeder.
- 6) Import: Farklı formatlarda dosya okur.
- 7) Page Setup: Sayfa ayarları yapar.
- 8) Plotter Setup: Yazıcı ayarları yapar.
- 9) Plot Preview: Çıktı ön izlemesi yapar.
- 10) Plot: Yazıcıdan çıktı alır.
- 11) Plot by Tiling: Parçalı baskı yapar.

New, open, save, save as, page setup, plotter setup, plot ve exit standart komutlardır. Import ve export komutları eklenmiştir.



Şekil 5.3 File menüsü

5.3.1.2 Edit

Edit menüsü şu komutlardan oluşmaktadır (Şekil 5.4):

- 1) Undo: Son yapılan işlemi geri alır.
- 2) Redo: Geri alınan işlemi iptal eder.
- 3) Cut: Seçilen elemanları kopyalar ve siler.
- 4) Copy: Seçilen elemanları kopyalar.
- 5) Paste: Kopyalanan elemanları yapıştırır.
- 6) Paste to Original Location: Kopyalanan elemanları orijinal koordinatlara yerleştirir.
- 7) Paste As Group: Kopyalanan elemanları gruplayarak yapıştırır.
- 8) Paste to New Story: Kopyalanan elemanları yeni bir kat planına yapıştırır.
- 9) Select All: Tüm elemanları seçer.
- 10) Deselect All: Seçili elemanları bırakır.
- 11) Invert Selection: Seçimi ters çevirir.
- 12) Delete: Seçili elemanları siler.
- 13) Clear: Tüm elemanları siler:
- 14) Group: Seçili elemanları gruplar.

15) Ungroup: Seçili grupları bozar.

Undo, redo, cut, copy, paste, select all ve delete standart komutlardır.

Undo	Ctrl+Z
Redo	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Paste To Original Location	
Paste As Group	
Paste To New Story	
SelectAll	Ctrl+A
DeselectAll	
InvertSelection	
Delete	Del
Clear	
Group	Ctrl+G
UnGroup	Shift+Ctrl+G

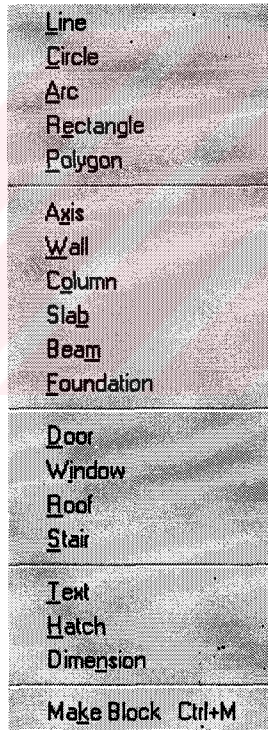
Şekil 5.4 Edit menüsü

5.3.1.3 Draw

Draw menüsü şu komutlardan oluşmaktadır (Şekil 5.4):

- 1) Line: Çizgi çizer.
- 2) Circle: Çember çizer.
- 3) Arc: Yay çizer.
- 4) Rectangle: Dikdörtgen çizer.
- 5) Polygon: Poligon çizer.
- 6) Axis: Aks çizer.
- 7) Wall: Duvar çizer.
- 8) Column: Kolon çizer.
- 9) Slab: Döşeme çizer.
- 10) Beam: Kiriş çizer.

- 11) Foundation: Temel çizer.
- 12) Door: Kapı çizer.
- 13) Window: Pencere çizer.
- 14) Roof: Çatı çizer.
- 15) Stair: Merdiven çizer.
- 16) Text: Yazı yazar.
- 17) Hatch: Tarama yapar.
- 18) Dimension: Ölçü oluşturur.
- 19) Make Block: Blok Yapar.



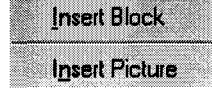
Şekil 5.5 Draw menüsü

5.3.1.4 Insert

Insert menüsü şu komutlardan oluşmaktadır (Şekil 5.6):

- 1) Insert Block: Dosya veya blok ekler.

2) Insert Picture: Çizime resim ekler.

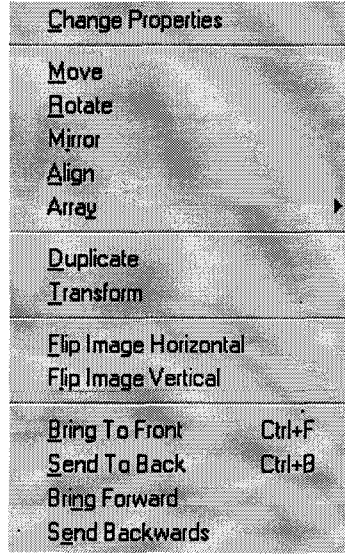


Şekil 5.6 Insert menüsü

5.3.1.5 Modify

Modify menüsü şu komutlardan oluşmaktadır (Şekil 5.7):

- 1) Change Properties: Seçilen elemanın özelliklerini değiştirir.
- 2) Move: Seçilen elemanı taşır.
- 3) Rotate: Seçilen elemanı döndürür.
- 4) Mirror: Seçilen elemanı aynalar.
- 5) Align: Seçilen elemanları hizalar.
- 6) Array: Seçilen elemanları doğrultu üzerinde dizer.
- 7) Duplicate: Seçilen elemanları çoğaltır.
- 8) Flip Image Horizontal: Seçilen resmi yatay olarak çevirir.
- 9) Flip Image Vertical: Seçilen resmi dikey olarak çevirir.
- 10) Bring to Front: Seçilen elemanı öne çıkartır.
- 11) Send to Back: Seçilen elemanı bir basamak geri plana atar.
- 12) Bring Forward: Seçilen elemanı bir basamak ileri çıkartır.
- 13) Send Backwards: Seçilen elemanı en arkaya gönderir.

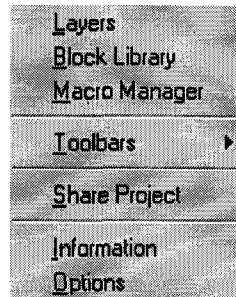


Şekil 5.7 Modify menüsü

5.3.1.6 Tools

Tools menüsü şu komutlardan oluşmaktadır (Şekil 5.8):

- 1) Layers: Katman yöneticisini açar.
- 2) Block Library: Blok kütüphanesini açar.
- 3) Macro Manager: Makro yöneticisini açar.
- 4) Toolbars: Araç çubuklarını gösterir.
- 5) Share Project: Projeyi Internet ve LAN üzerinden paylaşır.
- 6) Information: Program ile ilgili bilgi verir.
- 7) Options: Program ayarlarını gösterir.

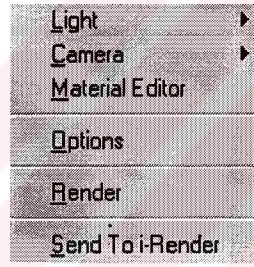


Şekil 5.8 Tools menüsü

5.3.1.7 Render

Render menüsündeki komutlar şunlardır (Şekil 5.9):

- 1) Light: Işık ayarları.
- 2) Camera: Kamera ayarları.
- 3) Material Editor: Malzeme ayarları.
- 4) Options: Render özellikleri.
- 5) Render: Render.
- 6) Send to i-Render: i-Render adlı render yazılımına proje yollar.

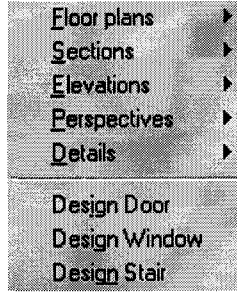


Şekil 5.9 Render menüsü

5.3.1.8 Design Studio

Design Studio menüsü içindeki komutlar şunlardır (Şekil 5.10):

- 1) Floor Plans: Kat ayarları.
- 2) Sections: Kesit ayarları
- 3) Elevations: Görünüş ayarları
- 4) Perspectives: Perspektif ayarları
- 5) Details: Detay ayarları
- 6) Design Door: Kapı tasarımları.
- 7) Design Stair: Merdiven tasarımları.

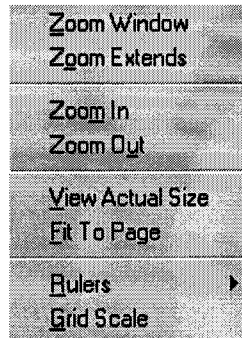


Şekil 5.10 Design Studio menüsü

5.3.1.9 View

View menüsündeki komutlar şunlardır (Şekil 5.11):

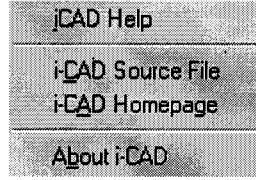
- 1) Zoom Window: Pencere ile seçilen alanı ekrana sığdırır.
- 2) Zoom Extends: Çizimi ekrana sığdırır.
- 3) Zoom In: Çizimi yakınlaştırır.
- 4) Zoom Out: Çizimi uzaklaştırır.
- 5) View Actual Size: Çizimi gerçek boyutta gösterir.
- 6) Fit to Page: Çizimi ekrana sığdırır.
- 7) Rulers: Cetvel ayarları.
- 8) Grid Scale: Izgara ayarları.



Şekil 5.11 View menüsü

5.3.1.10 Help

- 1) i-CAD Help: i-CAD yardım dosyası.
- 2) i-CAD Source File: i-CAD kaynak kodunu açar.
- 3) i-CAD Homepage: i-CAD Web sitesine bağlanır.
- 4) About i-CAD: i-CAD hakkında bilgi verir.



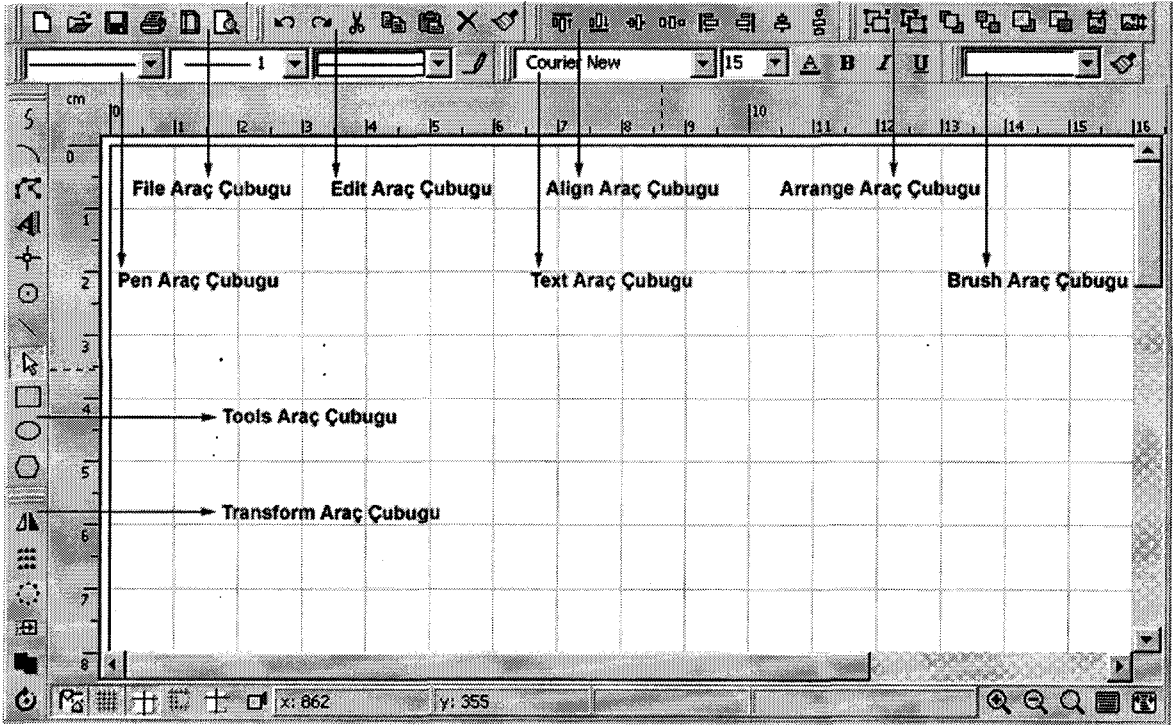
Şekil 5.12 Help menüsü

5.3.2 Araç Çubukları

Araç çubukları dokuz grupta toplanmıştır (Şekil 5.13):

- 1) File
- 2) Transform
- 3) Edit
- 4) Text
- 5) Pen
- 6) Brush
- 7) Align
- 8) Arrange
- 9) Tools

Araç çubuklarının yerleri değiştirilebilir ve yeni araç çubukları yapılabilir. Gruplardaki komutlara ekleme veya çıkarma yapılabilir.

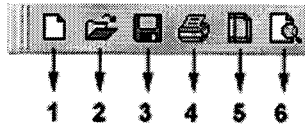


Şekil 5.13 Araç çubukları

5.3.2.1 File

File araç çubuğu şu komutlardan oluşmaktadır (Şekil 5.14):

- 1) New: Yeni dosya açar.
- 2) Open: Dosya açar.
- 3) Save: Dosya kaydeder.
- 4) Plot: Yazıcıdan çıktı alır.
- 5) Plotter Setup: Yazıcı ayarı yapar.
- 6) Plot Preview: Çıktı ön izlemesi yapar.

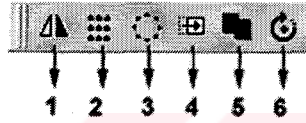


Şekil 5.14 File Araç çubuğu

5.3.2.2 Transform

Transform araç çubuğu şu komutlardan oluşmaktadır (Şekil 5.15):

- 1) Mirror: Seçilen elemanı aynalar.
- 2) Rectangular Array: Seçilen elemanları dikdörtgen doğrultu üzerinde dizer.
- 3) Polar Array: Seçilen elemanları daire üzerinde dizer.
- 4) Move: Seçilen elemanı taşır.
- 5) Copy: Seçilen elemanları kopyalar
- 6) Rotate: Seçilen elemanı döndürür.

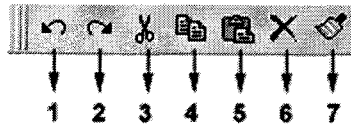


Şekil 5.15 Transform araç çubuğu

5.3.2.3 Edit

Edit araç çubuğu şu komutlardan oluşmaktadır (Şekil 5.16):

- 1) Undo: Son yapılan işlemi geri alır.
- 2) Redo: Geri alınan işlemi iptal eder.
- 3) Cut: Seçilen elemanları kopyalar ve siler.
- 4) Copy: Seçilen elemanları kopyalar.
- 5) Paste: Kopyalanan elemanları yapıştırır.
- 6) Delete: Seçili elemanları siler.
- 7) Clear: Tüm elemanları siler

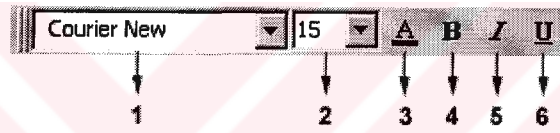


Şekil 5.16 Edit Araç çubuğu

5.3.2.4 Text

Text araç menüsü şu komutlardan oluşmaktadır (Şekil 5.17):

- 1) Font Type: Font tipi.
- 2) Font Size: Yazı büyüklüğü.
- 3) Font Color: Yazı rengi.
- 4) Bold Text: Kalın yazı.
- 5) Italic Text: Yatık yazı.
- 6) Underlined Text: Altı çizili yazı.

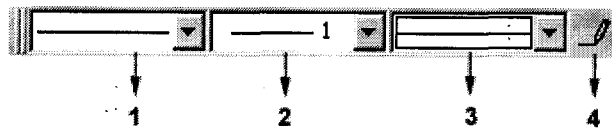


Şekil 5.17 Text araç menüsü

5.3.2.5 Pen

Pen araç menüsü şu komutlardan oluşmaktadır (Şekil 5.18):

- 1) Line Type: Çizgi tipi.
- 2) Line Width: Çizgi kalınlığı.
- 3) Line End Style: Çizgi uçları.
- 4) Line Color: Çizgi rengi.

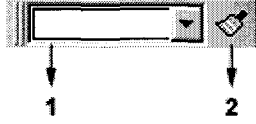


Şekil 5.18 Pen araç menüsü

5.3.2.6 Brush

Brush araç menüsü şu komutlardan oluşmaktadır (Şekil 5.19):

- 1) Hatch Type: Tarama tipi.
- 2) Hatch Color: Tarama rengi.

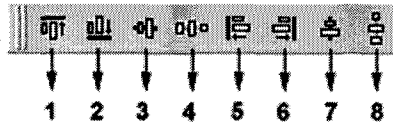


Şekil 5.19 Brush araç menüsü

5.3.2.7 Align

Align araç menüsü şu komutlardan oluşmaktadır (Şekil 5.20):

- 1) Align Top: Seçilen elemanları yukarıya doğru hizalar.
- 2) Align Bottom: Seçilen elemanları aşağıya doğru hizalar.
- 3) Distribute Horizontal Widths: Seçilen elemanları yatayda eşit aralıklar ile hizalar.
- 4) Align Horizontal Centers: Seçilen elemanları yatay merkeze göre hizalar.
- 5) Align Left: Seçilen elemanları sola doğru hizalar.
- 6) Align Right: Seçilen elemanları sağa doğru hizalar.
- 7) Distribute Vertical Widths: Seçilen elemanları düşeyde eşit aralıklar ile hizalar.
- 8) Align Vertical Centers: Seçilen elemanları düşey merkeze göre hizalar.

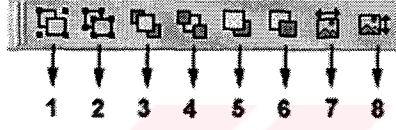


Şekil 5.20 Align araç menüsü

5.3.2.8 Arrange

Arrange araç menüsü şu komutlardan oluşmaktadır (Şekil 5.21):

- 1) Group: Seçili elemanları gruplar.
- 2) Ungroup: Seçili grupları bozar.
- 3) Bring to Front: Seçilen elemanı öne çıkarır.
- 4) Send to Back: Seçilen elemanı bir basamak geri plana atar.
- 5) Bring Forward: Seçilen elemanı bir basamak ileri çıkarır.
- 6) Send Backwards: Seçilen elemanı en arkaya gönderir.
- 7) Flip Image Horizontal: Seçilen resmi yatay olarak çevirir.
- 8) Flip Image Vertical: Seçilen resmi dikey olarak çevirir.



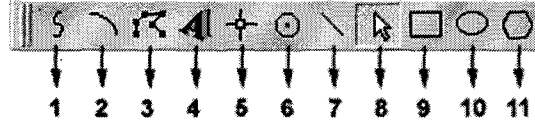
Şekil 5.21 Arrange araç menüsü

5.3.2.9 Tools

Tools araç menüsü şu komutlardan oluşmaktadır (Şekil 5.22):

- 1) Spline: Eğrisel çizgi çizer.
- 2) Arc: Yay çizer.
- 3) Polyline: Çoklu çizgi çizer.
- 4) Text: Yazı yazar.
- 5) Point: Nokta çizer.
- 6) Circle: Çember çizer.
- 7) Line: Çizgi çizer.
- 8) Select: Eleman seçer.
- 9) Rectangle: Dikdörtgen çizer.
- 10) Ellipse: Elips çizer.

11) Polygon: Poligon çizer.

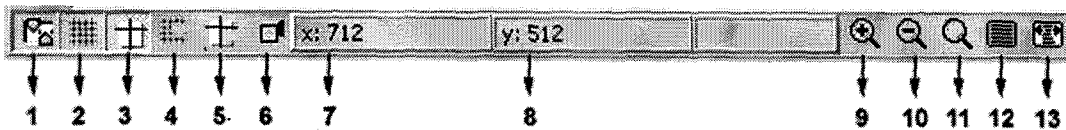


Şekil 5.22 Tools araç menüsü

5.3.3 Durum Çubuğu

Durum çubuğu şu komutlardan oluşmaktadır (Şekil 5.23):

- 1) View Rulers: Cetvelleri açma / kapama.
- 2) View Grid: Izgaraları açma / kapama.
- 3) View Guides: Kılavuz çizgileri açma / kapama.
- 4) Snap to Grids: Izgara noktalarını yakalama.
- 5) Snap to Guides: Klavuz çizgilerini yakalama.
- 6) Snap to Near Point: En yakın noktayı yakalama.
- 7) x Coordinate Box: x Koordinat kutusu.
- 8) y Coordinate Box: y Koordinat kutusu.
- 9) Zoom In: Çizim yakınlaştırır.
- 10) Zoom Out: Çizimi uzaklaştırır.
- 11) Zoom Window: Pencere ile seçilen alanı ekrana sığdırır.
- 12) View Actual Size: Çizimi gerçek boyutta gösterir.
- 13) Zoom Extends: Çizimi ekrana sığdırır.



Şekil 5.23 Durum çubuğu

5.3.4 Modify Araç Menüsü

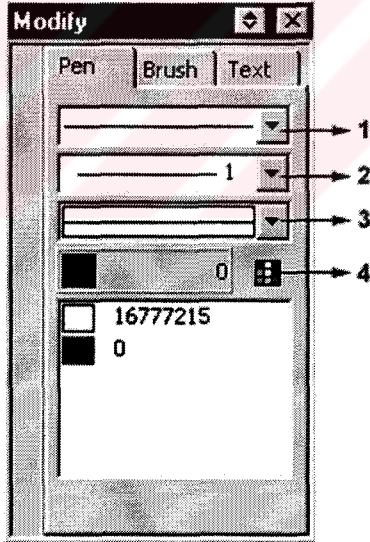
Modify Araç menüsü şu komutlardan oluşmaktadır:

- 1) Modify Pen
- 2) Modify Brush
- 3) Modify Text

5.3.4.1 Modify Pen

Modify Pen araç menüsü şu komutlardan oluşmaktadır (Şekil 5.24):

- 1) Line Type: Çizgi tipi.
- 2) Line Width: Çizgi kalınlığı.
- 3) Line End Style: Çizgi uçları.
- 4) Line Color: Çizgi rengi.



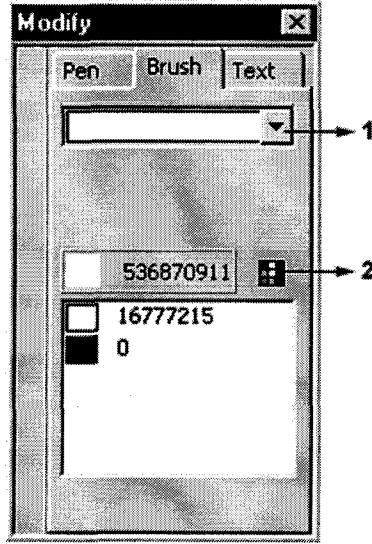
Şekil 5.24 Modify Pen araç menüsü

5.3.4.2 Modify Brush

Modify Brush araç menüsü şu komutlardan oluşmaktadır (Şekil 5.25):

- 1) Hatch Type: Tarama tipi.

2) Hatch Color: Tarama rengi.

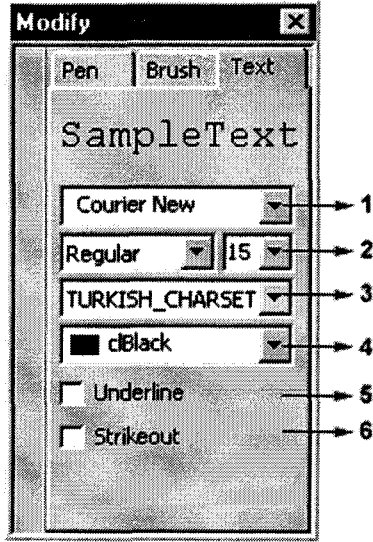


Şekil 5.25 Modify Brush araç menüsü

5.3.4.3 Modify Text

Modify Text araç menüsü şu komutlardan oluşmaktadır (Şekil 5.26):

- 1) Font Type: Font tipi.
- 2) Font Size: Yazı büyüklüğü.
- 3) Character Set: Yazı dil seti.
- 4) Font Color: Yazı rengi.
- 5) Underlined Text: Altı çizili yazı.
- 6) Strikeout Text: Üzeri çizili yazı.



Şekil 5.26 Modify Text araç menüsü

Bu bölümde kavramsal bir mimari yazılım arayüzü önerilmiştir. Bir ihtiyaç listesi oluşturularak, i-CAD adında bir yazılım geliştirilmiştir. Arayüz kuralları ile bir grafik arayüz tasarlanmıştır. Ancak bu kavramsal model, gerekli bilgisayar kodları olmadığı için çalışan bir yazılım konumunda değildir.

6. SONUÇ

Bilgisayarın, tanımlanma süreci, insanlık tarihi kadar eskidir. Bilgisayarın varolmasının ana sebebi ise sayı saymadır. Tarih boyunca hesap yapmayı kolaylaştıran yöntemler ve aletler icat edilmiştir. Bu yöntemler ve aletler, gelişen düşünceler ile birleştiğinde, bilgisayarın tanımı yapılabilmektedir, üretimine ise ancak 1950'li yıllarda başlanmıştır. İlk bilgisayarın üretiminden itibaren, kullanım problemleri de ortaya çıkmıştır. Bu problemler, İnsan-Bilgisayar Etkileşimi adı altında yeni araştırma alanının ortaya çıkmasına neden olmuştur.

İnsan-Bilgisayar Etkileşimi (İBE) alanında yapılan araştırmalar, bilgisayarların kullanımını kolaylaştırmış ve yaygınlaştırmıştır. Bilgisayarın kullanımının yaygınlaşmasıyla beraber, bütün meslek alanlarının çalışma metotları değişime uğramıştır. Tasarımla uğraşan meslekler de bu değişimden etkilenmiştir.

Önceleri üretim için kullanılan bilgisayarların, teknolojinin gelişmesiyle beraber, tasarım sürecinde de kullanılması olanağı doğmuştur. Günümüzde, bilgisayar teknolojileri, mimarlık mesleğinin sürdürülebilirliği biçimini etkilemektedir. Bilgisayar destekli tasarım süreci, mimarlık mesleğinde yeni ufuklar açmıştır.

Tasarım sürecinde bilgisayar çeşitli alanlarda kullanılmaktadır. Bu alanlarda farklı yazılımlar geliştirilmektedir. Bu yazılımlar birbirlerinden kopuk olarak tasarlanmakta ve mimari düşünce şekline uygun olmadıklarından dolayı yetersiz kalmaktadırlar. Bilgisayar destekli tasarım yazılımları tasarım sürecinin tamamında kullanılamamaktadır. Bilgisayar ortamında model oluşturma süreci uzun ve zordur. Bilgisayarda model oluşturulduktan sonra, simülasyon yapımı için ise yeterli yazılım bulunmaktadır.

Bilgisayar destekli tasarım alanında kullanılan yazılımlar yeterli ölçüde ortak çalışma yapılmasına olanak sağlamamaktadır. Ortak çalışma yapmak için sanal tasarım stüdyoları kullanımı ise yeni bir oluşumdur. Tasarım sürecinde henüz yeterli kullanım alanı bulamamıştır.

Mimari tasarım için kullanılacak bir yazılımın modelleme, simülasyon ve ortak çalışma gibi özellikleri bir arada bulundurması gerekmektedir. Bu alanda kullanılacak bir yazılım, bütün bu özellikleri kapsayan bir ihtiyaç listesine göre tasarlanmalıdır. Bu ihtiyaç listesi mimari çalışma yöntemi ile birleştirilip bir arayüz geliştirilmelidir. Günümüzde kullanılan yazılımlar ise bu şekilde geliştirilmemektedir. Bu çalışmada önerilen modelde böyle bir yöntem denenmiştir. Gelecekte bu yöndeki çalışmaların artacağı düşünülmektedir.

Bu arařtırmada, mimari tasarıma yönelik bir BDT (Bilgisayar Destekli Tasarım) yazılımı geliřtirilmeye alıřılmıřtır. alıřmada kullanılan programlama bilgisi, gerek bir BDT yazılımı üretimine imkan vermemektedir. Bu yzden bu model kavramsal dzeyde kalmıřtır. Modelin ana hatlarını oluřturan ihtiya listesi, kullanılmakta olan yazılımlar erevesinde genel olarak belirlenmiřtir. Belirlenen ihtiya listesi, kavramsal bir modelin geliřtirilmesi iin yeterlidir, ancak alıřan bir model iin daha kapsamlı bir ihtiya listesi oluřturulmalıdır.

Bu kavramsal modelin arayz, Delphi IDE (Integrated Development Environment) ile yapılmıřtır, bununla beraber herhangi bir bařka IDE yazılımı da kullanmak mmkndr. Bu kavramsal modelin kaynak kodları, GNU GPL (GNU General Public License/ GNU Genel Kamu Lisansı) lisansı ile daėıtılmaktadır. Bylelikle, bu model, konu zerinde alıřanlar tarafından geliřtirilebilir. Oluřturulan modelin kavramsal olması sebebi ile, kullanıcılar zerinde test yapılamamıřtır. Bu nerinin rne dnřtrlmesi ve test edilmesinin daha kapsamlı bir alıřma konusu olarak, ilerde ele alınması dřnlmektedir.



KAYNAKLAR

- Baecker, R.M., (1993), Readings in Groupware and Computer Supported Cooperative Work, Morgan Kaufmann Publishers, ABD.
- Dumas, J., Redish, J., (1993), Practical Guide for Usability Testing, Ablex Publishing Corporation, Norwood, NJ
- Edwards, P., (1996), The Closed World, MIT Press, Cambridge, Massachusetts
- Ifrah, G., (2000), Hesabın Destanı: Rakamların Evrensel Tarihi, Tubitak Popüler Bilim Kitapları, İstanbul
- Jones, Mark., (1998), Human Computer Interaction: A Design Guide, Educational Technology Publications, NJ
- Kistermann, F. W., (1998), The Tabulator DEHOMAG D 11, June 1998, Paderborn
- Kurzweil, R., (1992), The age of Intelligent Machines, MIT Press, Cambridge, Massachusetts
- Kurzweil, R., (1998), The age of spiritual Machines, 1998 MIT Press, Cambridge, Massachusetts
- Laiserin, J., (2000), Challenges for the Digital Generation, Architectural Record, 188/12, 166-169, New York
- Laurel, B., (1990), The art Human Computer Interface Design, Addison Wesley Co, Massachusetts
- Lindgaard, G., (1994), Usability Testing System Evaluation, Chapman & Hall, London
- Lindgren, M., (1996), Glory and Failure, MIT Press, Cambridge, Massachusetts
- Nielsen, J., (1994), Usability Engineering, Academic Press, Boston
- Norman, D. A., (1990), The Design of Everyday Things, Doubleday
- Novitski, B. J., (1999), Digital Architect, Architectural Record, 187-2, 43-44, New York
- Picturetel 900 Series Collaboration System, (2000), Picturetel Corporation
- Preece, J., (1994), Human Computer Interaction, Addison-Wesley Publishing Company, Massachusetts
- Rheingold, H., (1985), Tools For Thought: The People and Ideas of the Next Computer Revolution, Simon & Schuster
- Sarıyar, A., (2001), Bilgisayar Destekli İletişim ve Öğrenim Amaçlı Mimari Tasarım Stüdyosunda Yeni Etkileşim Olanakları, YTÜ Mimarlık Fakültesi Mimarlık Bölümü Yüksek Lisans Tezi, İstanbul.
- Shneiderman, B., (1998), Designing the User Interface, Addison-Wesley Publishing Company, California
- Türkoğlu, T., (2001), Delphi Programcılığı ve SQL, Çayyolu, Ankara
- Walker, J., (1994), The Autodesk File: Bits of History Words of Experience, New Riders Publishing, NY

Williams S., Walmsley, S., (1998), Discover Delphi, Pearson Education Limited, Essex

Wilson, P., (1991) An Introduction to CSCW, Kluwer Academic Publishers, Netherlands.

Winograd, T., (1996), Bringing Design to Software, Addison-Wesley, CA

Wojtowics, J., Davidson, J N., ve Nagakura, T (1995), "Digital Pinup Board- the story of the virtual village Project," Virtual Design Studio, Hong Kong University Press, Hong Kong



INTERNET KAYNAKLARI

- (1) Alan Turing Home Page, www.turmg.org.uk
- (2) Allison, D., Using The Computer: Episodes across 50 years, A Presentation to the Annual Meeting Association of Computing Machinery, (February 14, 1996), Smithsonian Institution
- (3) Ambler, S., User Interface Design, (2000), <http://www.ambysoft.com/userInterfaceDesign.pdf>
- (4) Autodesk, www.autodesk.com
- (5) Bentley Systems, www.bentley.com
- (6) Brennecke, A., A Classification Scheme for Program Controlled Calculators, Introduction to the International Conference on the History of Computing, 14. -16. August 1998, Paderborn, www.inf.fu-berlin.de/~widiger/ICHC/papers
- (7) BricsNet, www.bricsnet.com
- (8) Carbons to Computers, <http://educate.si.edu/scitech/carbons/index.htm>
- (9) Chronology of Personal Computers, <http://www.islandnet.com/~kpolsson>
- (10) Cinema4D, www.cinema4d.com
- (11) Computer Hope, www.computerhope.com
- (12) Computers Society, Events in the history of computing, www.computer.org/history/development/index.html
- (13) Denholm, S., Computer Aided Design, <http://www.personal.dundee.ac.uk/~mgzforbe/it100/report/report.html>
- (14) Graphisoft Inc., www.graphisoft.com
- (15) Green, E., (1999), Xerox PARC, <http://cc.kzoo.edu/~k98ab01/>
- (16) Gustafson, The reconstruction of Atanasoff Berry Computer, Introduction to the International Conference on the History of Computing, 14. -16. August 1998, Paderborn, www.inf.fu-berlin.de/~widiger/ICHC/papers
- (17) Hardy, C., Human Computer Interaction, (1998), http://osiris.sunderland.ac.uk/~cs0car/hci/hci_all.htm
- (18) Help&Manual, www.ec-software.com
- (19) History of Windows API, <http://www.iseran.com/>
- (20) IBM, www.ibm.com
- (21) I-MB CAD Chronology, <http://www.bozdoc.f2s.com>
- (22) Installshield, www.installshield.com/express
- (23) İde Yapı, www.idecad.com

- (24) Jones International and Jones Digital Century, Computers: History and Development, www.digitalcentury.com/encyclo/index.html
- (25) Kempf, K., Electronic Computers Within The Ordnance Corps, <http://ftp.arl.army.mil/~mike/comphist/61ordnance>
- (26) Kinetix, www.kinetix.com
- (27) Lee, J., John Louis von Neumann, (1997), <http://ei.cs.vt.edu/~history/VonNeumann.html>
- (28) Max Frame Corporation, History of Computing, www.maxframe.com
- (29) Microsoft Developer Network, <http://msdn.microsoft.com>
- (30) Microsoft MSN, www.msn.com
- (31) Microsoft, www.microsoft.com
- (32) Mirabilis Software, www.icq.com
- (33) Multics Internet sitesi, www.multicians.org
- (34) Nemetschek AG, www.nemetschek.com
- (35) PcMuseum, <http://members.fortunecity.com/pcmuseum>
- (36) Personal Computers: History and Development, www.digitalcentury.com
- (37) PictureTel, www.picturetel.com
- (38) Preston, V., Creating Value for Xerox and the World, <http://cc.kzoo.edu/~k97sm03/>
- (39) Primavera, www.primavera.com
- (40) Rojas, R., How to Make Zuse's Z3 a Universal Computer, (1998), Introduction to the International Conference on the History of Computing, 14. -16. August 1998, Paderborn, www.inf.fu-berlin.de/~widiger/ICHC/papers
- (41) Sap2000, www.csiberkeley.com
- (42) Shehata, C., Language Usage in Design, <http://www.shehata.com/HCI>
- (43) Smith, S., Urban Simulation: Cities of the Future, A/E/C SYSTEMS: The Magazine of Computer Solutions, www.penton.com
- (44) Superonline, www.superonline.com
- (45) Tekhnologos Software, www.tekhnologos.com
- (46) Tekla XSteel, www.xsteel.com
- (47) The History of Computer Foundation, www.thocp.net/index.htm
- (48) The History of Computing at BRL, <http://ftp.arl.army.mil/ftp/historic-computers>
- (49) The Unix System: History and Timeline, www.unix-systems.org
- (50) Time Warner American Online, www.aol.com

- (51) Türk Bilişim Derneği, www.bdt.org.tr
- (52) White, S., A Brief history Of Computing, (1996-2001),
<http://ox.compsoc.net/~swhite/history/history.html>
- (53) Wikipedia Operating Systems, www.wikipedia.com
- (54) Williams, M. R., A Preview of Things to come: Some remarks on the First Generation Computers, <http://www.inf.fu-berlin.de/~widiger/ICHC/papers/Williams.html>



EKLER

Ek 1: i-cad.dpr Delphi 5 dosyası

Ek 2: main.pas Delphi 5 dosyası

Ek 3: about.pas Delphi 5 dosyası

Ek 4: i-CAD.iwz Installshield Express dosyası

Ek 5: i-CAD CD'si (i-CAD "kur" dosyası, yardım dosyaları, program kodları)



Ek 1: i-cad.dpr Delphi 5 dosyası

```
program iCAD;

uses
  Forms,
  main in 'main.pas' {Form1},
  about in 'about.pas' {frmAbout};

{$R *.RES}

begin
  Application.Initialize;
  Application.Title := 'i-CAD';
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TfrmAbout, frmAbout);
  Application.Run;
end.
```



Ek 2: main.pas Delphi 5 dosyası

```

(*****
* User Interface Model for Computer Aided Architectural Design
* Copyright (C) 2002 Emrah Turkyilmaz
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
* -----
* I hope You send me a copy of any important changes/additions You do.
*
* E-mail: e.turkyilmaz@iku.edu.tr
*
*****)

```

```
unit main;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, PCPanel, PCDrawing, PowerCad, PCAlignBar,
PCTransformBar, PCMSBar, BarBase, PCToolsBar, Menus,
PCTransformDlg, PCPageDlg, PCOptionsDlg, PCLayerDlg,
DlgBase, PCAlignDlg, PCTypesUtils, PCModifyDlg, PCBlockDlg, DrawObjects,
PCModifyBar, PCMacroDlg, ExtDlgs, PCFileDlgs, PCDrawBox, PCArrangeBar,
PCCommandBar, PCEditBar, PCFileBar, OleCtrls,
StdCtrls, printers, ActnList, DrawEngine, OleCtrls, SHDocVw, NMURL,
ehsHelpRouter, BULabel, BUHotLink, BUExecute;
```

```
type
```

```
TForm1 = class(TForm)
  DockTop: TPCDock;
  DockRight: TPCDock;
  DockLeft: TPCDock;
  DockBottom: TPCDock;
  MainMenu1: TMainMenu;
  File1: TMenuItem;
  iCAD: TPowerCad;
  Open1: TMenuItem;
  Savel: TMenuItem;
  SaveAs1: TMenuItem;
  Exit1: TMenuItem;
  iCADAlignDlg1: TPCAlignDlg;
  iCADLayerDlg1: TPCLayerDlg;
  iCADOptionsDlg1: TPCOptionsDlg;
  iCADPageDlg1: TPCPageDlg;
  iCADTransformDlg1: TPCTransformDlg;
  Edit1: TMenuItem;
  SelectAll1: TMenuItem;
  DeselectAll1: TMenuItem;
end;
```

```
InvertSelection1: TMenuItem;
N2: TMenuItem;
Copy1: TMenuItem;
Cut1: TMenuItem;
Paste1: TMenuItem;
N3: TMenuItem;
Delete1: TMenuItem;
Clear1: TMenuItem;
Undo1: TMenuItem;
N4: TMenuItem;
Arrange1: TMenuItem;
Help1: TMenuItem;
N11: TMenuItem;
Register1: TMenuItem;
iCADTransformBar1: TPCTransformBar;
iCADModifyDlg1: TPCModifyDlg;
iCADBlockDlg1: TPCBlockDlg;
iCADModifyBar1: TPCModifyBar;
iCADModifyBar2: TPCModifyBar;
iCADModifyBar3: TPCModifyBar;
iCADOpenDialog1: TPCOpenDialog;
iCADSaveDialog1: TPCSaveDialog;
New1: TMenuItem;
Print1: TMenuItem;
N15: TMenuItem;
PrintDialog1: TPrintDialog;
SaveDialog1: TSaveDialog;
PrinterSetupDialog1: TPrinterSetupDialog;
iCADFileBar1: TPCFileBar;
PrinterSetup1: TMenuItem;
PrintPreview1: TMenuItem;
iCADEditBar1: TPCeditBar;
N16: TMenuItem;
Dialogs1: TMenuItem;
Options2: TMenuItem;
N20: TMenuItem;
BlockLibrary1: TMenuItem;
Layers1: TMenuItem;
OpenPictureDialog2: TOpenPictureDialog;
InsertBlock1: TMenuItem;
InsertPicture1: TMenuItem;
iCADArrangeBar1: TPCArrangeBar;
iCADToolsBar1: TPCToolsBar;
iCADAlignBar1: TPCAlignBar;
DXFDial: TOpenDialog;
Information1: TMenuItem;
PrintByTiling1: TMenuItem;
iCADMacroDialog1: TPCMacroDialog;
N23: TMenuItem;
N24: TMenuItem;
Redo1: TMenuItem;
PasteToOriginalLocation1: TMenuItem;
PasteasGroup1: TMenuItem;
PasteToNewStory1: TMenuItem;
N13: TMenuItem;
N25: TMenuItem;
Group1: TMenuItem;
UnGroup1: TMenuItem;
N5: TMenuItem;
Draw1: TMenuItem;
Modify2: TMenuItem;
FlipImageHorizontal1: TMenuItem;
```

```
FlipImageVertical1: TMenuItem;
N1: TMenuItem;
BringToFront1: TMenuItem;
SendToBack1: TMenuItem;
BringForward1: TMenuItem;
SendBackwards1: TMenuItem;
MakeBlock1: TMenuItem;
N12: TMenuItem;
N14: TMenuItem;
Move1: TMenuItem;
Rotate1: TMenuItem;
Array1: TMenuItem;
Rectangular1: TMenuItem;
Polar1: TMenuItem;
Mirror1: TMenuItem;
Duplicate1: TMenuItem;
PageSetup1: TMenuItem;
MacroManager1: TMenuItem;
Modify1: TMenuItem;
Align1: TMenuItem;
Transform1: TMenuItem;
N7: TMenuItem;
View1: TMenuItem;
Zoom1: TMenuItem;
ZoomExtends1: TMenuItem;
ZoomIn1: TMenuItem;
ZoomOut1: TMenuItem;
ViewActualSize1: TMenuItem;
FitToPage1: TMenuItem;
N6: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
Rulers1: TMenuItem;
HideRulers1: TMenuItem;
ViewRulers1: TMenuItem;
RulerUnit1: TMenuItem;
Metric1: TMenuItem;
Imperial1: TMenuItem;
GridScale1: TMenuItem;
Toolbars1: TMenuItem;
File2: TMenuItem;
Tools1: TMenuItem;
Transform2: TMenuItem;
N17: TMenuItem;
Edit2: TMenuItem;
Text1: TMenuItem;
Pen1: TMenuItem;
Brush1: TMenuItem;
N18: TMenuItem;
Align2: TMenuItem;
Arrange2: TMenuItem;
N21: TMenuItem;
Import1: TMenuItem;
ImportDXF1: TMenuItem;
AutoCADDWG1: TMenuItem;
N3DStudio3DS1: TMenuItem;
Export1: TMenuItem;
ExportAs1: TMenuItem;
BMPFile1: TMenuItem;
DXFFile1: TMenuItem;
DWGFile1: TMenuItem;
Line1: TMenuItem;
```

```
Circle1: TMenuItem;
Rectangle1: TMenuItem;
Polygon1: TMenuItem;
Arc1: TMenuItem;
N19: TMenuItem;
Wall1: TMenuItem;
Column1: TMenuItem;
Beam1: TMenuItem;
Slab1: TMenuItem;
Stair1: TMenuItem;
Foundation1: TMenuItem;
N22: TMenuItem;
Axis1: TMenuItem;
Text2: TMenuItem;
Dimension1: TMenuItem;
Hatch1: TMenuItem;
N26: TMenuItem;
Door1: TMenuItem;
Window1: TMenuItem;
N27: TMenuItem;
ShareProject1: TMenuItem;
N28: TMenuItem;
Render1: TMenuItem;
MaterialEditor1: TMenuItem;
Light1: TMenuItem;
Camera1: TMenuItem;
N29: TMenuItem;
SendToRender1: TMenuItem;
N30: TMenuItem;
Options1: TMenuItem;
Render2: TMenuItem;
Create1: TMenuItem;
Edit3: TMenuItem;
Create2: TMenuItem;
Edit4: TMenuItem;
N31: TMenuItem;
DesignStudio1: TMenuItem;
Floorplans1: TMenuItem;
Sections1: TMenuItem;
Elevations1: TMenuItem;
Perspectives1: TMenuItem;
Details1: TMenuItem;
CreateNewFloor1: TMenuItem;
N1StoryFloorPlan1: TMenuItem;
View2: TMenuItem;
Edit5: TMenuItem;
N1StoryFloorPlan2: TMenuItem;
N32: TMenuItem;
Edit6: TMenuItem;
View3: TMenuItem;
N1StoryFloorPlan3: TMenuItem;
N33: TMenuItem;
Edit7: TMenuItem;
View4: TMenuItem;
N1StoryFloorPlan4: TMenuItem;
N34: TMenuItem;
Edit8: TMenuItem;
View5: TMenuItem;
CreateNewSection1: TMenuItem;
SectionAA1: TMenuItem;
View6: TMenuItem;
Edit9: TMenuItem;
```

```
SectionAA2: TMenuItem;
Edit10: TMenuItem;
View7: TMenuItem;
SectionAA3: TMenuItem;
Edit11: TMenuItem;
View8: TMenuItem;
CreateNewElevation1: TMenuItem;
NorthElevation1: TMenuItem;
View9: TMenuItem;
Edit12: TMenuItem;
NorthElevation2: TMenuItem;
Edit13: TMenuItem;
View10: TMenuItem;
NorthElevation3: TMenuItem;
Edit14: TMenuItem;
View11: TMenuItem;
NorthElevation4: TMenuItem;
Edit15: TMenuItem;
View12: TMenuItem;
CreateNewPerspective1: TMenuItem;
Perspective11: TMenuItem;
View13: TMenuItem;
Edit16: TMenuItem;
Perspective12: TMenuItem;
Edit17: TMenuItem;
View14: TMenuItem;
CreateNewDetail1: TMenuItem;
Perspective13: TMenuItem;
Edit18: TMenuItem;
View15: TMenuItem;
DetailA1: TMenuItem;
Edit19: TMenuItem;
View16: TMenuItem;
DetailA2: TMenuItem;
Edit20: TMenuItem;
View17: TMenuItem;
DetailA3: TMenuItem;
Edit21: TMenuItem;
View18: TMenuItem;
DetailA4: TMenuItem;
Edit22: TMenuItem;
View19: TMenuItem;
N35: TMenuItem;
DesignDoorWindow1: TMenuItem;
DesignStair1: TMenuItem;
Roof1: TMenuItem;
DesignWindow1: TMenuItem;
iCADSourceFile1: TMenuItem;
iCADHelp1: TMenuItem;
N36: TMenuItem;
iCADHelp: THelpRouter;
iCADHomepagel: TMenuItem;
Homepage: TBUHotLink;
SourceExecute: TBUExecute;
iCADHelpwin951: TMenuItem;
procedure Exit1Click(Sender: TObject);
procedure SaveAs1Click(Sender: TObject);
procedure Save1Click(Sender: TObject);
procedure SelectAll1Click(Sender: TObject);
procedure DeselectAll1Click(Sender: TObject);
procedure Delete1Click(Sender: TObject);
procedure Group1Click(Sender: TObject);
```

```

procedure UnGroup1Click(Sender: TObject);
procedure MakeBlock1Click(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Clear1Click(Sender: TObject);
procedure Copy1Click(Sender: TObject);
procedure Cut1Click(Sender: TObject);
procedure Paste1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure InvertArc1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure New1Click(Sender: TObject);
procedure ExportAs1Click(Sender: TObject);
procedure Print1Click(Sender: TObject);
procedure iCADFileBar1PrintClick(Sender: TObject);
procedure PrinterSetup1Click(Sender: TObject);
procedure FileBar1Click(Sender: TObject);
procedure ToolsBar1Click(Sender: TObject);
procedure TransformBar1Click(Sender: TObject);
procedure Editbar2Click(Sender: TObject);
procedure TextBar1Click(Sender: TObject);
procedure PenBar1Click(Sender: TObject);
procedure BrushBar1Click(Sender: TObject);
procedure AlignBar2Click(Sender: TObject);
procedure PageSetup2Click(Sender: TObject);
procedure Options2Click(Sender: TObject);
procedure Align2Click(Sender: TObject);
procedure BlockLibrary1Click(Sender: TObject);
procedure Layers1Click(Sender: TObject);
procedure Modify1Click(Sender: TObject);
procedure Transform2Click(Sender: TObject);
procedure Register1Click(Sender: TObject);
procedure Undo1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure PluginClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormDeactivate(Sender: TObject);
procedure InsertBlock1Click(Sender: TObject);
procedure InsertPicture1Click(Sender: TObject);
procedure Combine1Click(Sender: TObject);
procedure Break1Click(Sender: TObject);
procedure BringToFront2Click(Sender: TObject);
procedure SendToBack2Click(Sender: TObject);
procedure BringForward2Click(Sender: TObject);
procedure SendBackward1Click(Sender: TObject);
procedure ArcStyleNormal1Click(Sender: TObject);
procedure ArcStylePie1Click(Sender: TObject);
procedure ArcStyleChord1Click(Sender: TObject);
procedure ImportDXF1Click(Sender: TObject);
procedure iCADSelectionChange(Sender: TObject);
procedure PrintByTiling1Click(Sender: TObject);
procedure iCADHelp1Click(Sender: TObject);
procedure iCADHomepage1Click(Sender: TObject);
procedure iCADSourceFile1Click(Sender: TObject);
procedure iCADHelpwin951Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

```

```
var
```

```

Form1: TForm1;
filename : string;
var
  ForceCurrentDirectory: Boolean = False;
implementation

uses about;

{$R *.DFM}

procedure TForm1.Exit1Click(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TForm1.SaveAs1Click(Sender: TObject);
begin
  if iCADsavedialog1.Execute then
  begin
    filename := iCADSavedialog1.filename;
    if (iCADSaveDialog1.FilterIndex = 1) and
      (LowerCase(ExtractFileExt(filename)) <> '.cad' )
    then filename := filename + '.cad';

    iCAD.SaveToFile(0, filename);
    Form1.Caption := 'i-CAD - '+filename;
  end;
end;

procedure TForm1.Save1Click(Sender: TObject);
begin
  if filename = '' then
  begin
    if iCADsavedialog1.Execute then
    begin
      filename := iCADSavedialog1.filename;
      if (iCADSaveDialog1.FilterIndex = 1) and
        (LowerCase(ExtractFileExt(filename)) <> '.cad' )
      then filename := filename + '.cad';

      iCAD.SaveToFile(0, filename);
      Form1.Caption := 'i-CAD - '+filename;
    end;
  end
  else
    iCAD.SaveToFile(0, filename);
end;

procedure TForm1.SelectAll1Click(Sender: TObject);
begin
  iCAD.selectall(iCAD.ActiveLayer);
end;

procedure TForm1.DeselectAll1Click(Sender: TObject);
begin
  iCAD.deselectall(iCAD.ActiveLayer);
end;

procedure TForm1.Delete1Click(Sender: TObject);

```

```

begin
  iCAD.RemoveSelection;
end;

procedure TForm1.Group1Click(Sender: TObject);
begin
  iCAD.groupselection;
end;

procedure TForm1.UnGroup1Click(Sender: TObject);
begin
  iCAD.ungroupselection;
end;

procedure TForm1.MakeBlock1Click(Sender: TObject);
var blname : string;
begin
  if InputQuery('Make Block', 'Block Name.', blname) then
    iCAD.MakeSelectionBlock(iCAD.ActiveLayer, iCAD.BlockDirectory+blname+'.blk')
  ;
end;

procedure TForm1.Open1Click(Sender: TObject);
begin
  if iCADOpenDialog1.execute then
    begin
      iCAD.LoadFromFile(iCADOpenDialog1.filename);
      filename := iCADOpenDialog1.filename;
      Form1.Caption := 'i-CAD - '+filename;
    end;
end;

procedure TForm1.Clear1Click(Sender: TObject);
begin
  iCAD.Clear(iCAD.ActiveLayer);
end;

procedure TForm1.Copy1Click(Sender: TObject);
begin
  iCAD.CopyToClipboard;
end;

procedure TForm1.Cut1Click(Sender: TObject);
begin
  iCAD.CutToClipboard;
end;

procedure TForm1.Paste1Click(Sender: TObject);
begin
  iCAD.PasteFromClipboard(iCAD.ActiveLayer);
end;

procedure TForm1.FormShow(Sender: TObject);
begin
  iCADModifyDlg1.show;
end;

procedure TForm1.InvertArc1Click(Sender: TObject);
begin
  iCAD.InvertArcsOfSelection;
end;

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  iCAD.BlockDirectory := extractfilepath(application.ExeName)+'blocks\';
  iCAD.PluginDirectory := extractfilepath(application.ExeName)+'plugins\';

  if paramcount > 0 then
  begin
    iCAD.loadfromfile(paramstr(1));
    filename := paramstr(1);
    Form1.Caption := 'i-CAD - '+filename;
  end;
  iCAD.LoadPlugins;
end;

```

```

Procedure TForm1.PluginClick(sender: TObject);
var sub,par: TMenuItem;
Begin
  Sub := sender as TMenuItem;
  par := sub.Parent;
  iCAD.DoPluginVerb(par.tag,sub.tag);
End;

```

```

procedure TForm1.New1Click(Sender: TObject);
var ResWord: Word;
begin
  ResWord := messagedlg
    ('Do you want to save the changes',mtConfirmation,mbYesNoCancel,0);
  If ResWord = mrYes then
  begin
    if filename = '' then
    begin
      if iCADsavedialog1.Execute then
      begin
        filename := iCADSavedialog1.filename;
        if (iCADSaveDialog1.FilterIndex = 1) and
          (LowerCase(ExtractFileExt(filename)) <> '.cad' )
        then filename := filename + '.cad';

        iCAD.SaveToFile(0,filename);
        Form1.Caption := 'i-CAD - '+filename;
      end
      else
        exit;
    end
    else
      iCAD.SaveToFile(0,filename);
  end;
  If ResWord = MrCancel then Exit;
  iCAD.Clear(0);
  iCAD.DeleteAllUserLayers;
  iCAD.SaveState;
  FileName := '';
  Caption := 'i-CAD - Untitled';
end;

```

```

procedure TForm1.ExportAs1Click(Sender: TObject);
var exportname : string;
  ext: string;
begin
  SaveDialog1.Filter := 'Windows Meta File (*.wmf)|*.WMF|Windows Bitmap
(*.bmp)|*.BMP';

```

```

if Savedialog1.Execute then
begin
  exportname := Savedialog1.filename;
  ext := LowerCase(ExtractFileExt(Savedialog1.filename));
  if Savedialog1.FilterIndex = 1 then
  begin
    if lowercase(ext) <> '.wmf' then exportname := exportname +
'.wmf';
    iCAD.ExportAsWmf(exportname);
  end
  else if Savedialog1.FilterIndex = 2 then
  begin
    if lowercase(ext) <> '.bmp' then exportname := exportname + '.bmp';
    iCAD.SaveAsBitmap(exportname);
  end;
end;
end;

procedure TForm1.Print1Click(Sender: TObject);
begin
  if printdialog1.execute then
  begin
    iCAD.Print('i-CAD-'+extractfilename(filename));
  end;
end;

procedure TForm1.iCADFileBar1PrintClick(Sender: TObject);
begin
  iCAD.Print('i-CAD-'+extractfilename(filename));
end;

procedure TForm1.PrinterSetup1Click(Sender: TObject);
begin
  printersetupdialog1.Execute;
end;

procedure TForm1.FileBar1Click(Sender: TObject);
begin
  iCADFileBar1.visible := not iCADFileBar1.visible ;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.ToolsBar1Click(Sender: TObject);
begin
  iCADToolsBar1.visible := not iCADToolsBar1.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.TransformBar1Click(Sender: TObject);
begin
  iCADTransformBar1.visible := not iCADTransformBar1.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.Editbar2Click(Sender: TObject);
begin
  iCADEditBar1.visible := not iCADEditBar1.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.TextBar1Click(Sender: TObject);

```

```
begin
  iCADModifyBar2.visible := not iCADModifyBar2.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.PenBar1Click(Sender: TObject);
begin
  iCADModifyBar1.visible := not iCADModifyBar1.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.BrushBar1Click(Sender: TObject);
begin
  iCADModifyBar3.visible := not iCADModifyBar3.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.AlignBar2Click(Sender: TObject);
begin
  iCADAlignBar1.visible := not iCADAlignBar1.visible;
  (Sender as TMenuItem).checked := not ((Sender as TMenuItem).checked);
end;

procedure TForm1.PageSetup2Click(Sender: TObject);
begin
  iCADPageDlg1.show;
end;

procedure TForm1.Options2Click(Sender: TObject);
begin
  iCADOptionsDlg1.show;
end;

procedure TForm1.Align2Click(Sender: TObject);
begin
  iCADAlignDlg1.show;
end;

procedure TForm1.BlockLibrary1Click(Sender: TObject);
begin
  iCADBlockDlg1.show;
end;

procedure TForm1.Layers1Click(Sender: TObject);
begin
  iCADLayerDlg1.show;
end;

procedure TForm1.Modify1Click(Sender: TObject);
begin
  iCADModifyDlg1.show;
end;

procedure TForm1.Transform2Click(Sender: TObject);
begin
  iCADTransformDlg1.show;
end;

procedure TForm1.Register1Click(Sender: TObject);
begin
  frmAbout.showmodal;
end;
```

```

end;

procedure TForm1.Undo1Click(Sender: TObject);
begin
    iCAD.UnDo;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    iCAD.UnLoadPlugins;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    New1.ShortCut := ShortCut(word('N'), [ssCtrl]);
    Open1.ShortCut := ShortCut(word('O'), [ssCtrl]);
    Save1.ShortCut := ShortCut(word('S'), [ssCtrl]);
    Print1.ShortCut := ShortCut(word('P'), [ssCtrl]);
    Exit1.ShortCut := ShortCut(vk_F4, [ssAlt]);
    Undo1.ShortCut := ShortCut(word('Z'), [ssCtrl]);
    SelectAll1.ShortCut := ShortCut(word('A'), [ssCtrl]);
    Copy1.ShortCut := ShortCut(word('C'), [ssCtrl]);
    Cut1.ShortCut := ShortCut(word('X'), [ssCtrl]);
    Paste1.ShortCut := ShortCut(word('V'), [ssCtrl]);
    Delete1.ShortCut := ShortCut(vk_Delete, []);
    Group1.ShortCut := ShortCut(word('G'), [ssCtrl]);
    Ungroup1.ShortCut := ShortCut(word('G'), [ssCtrl, ssShift]);
    MakeBlock1.ShortCut := ShortCut(word('M'), [ssCtrl]);
    BringToFront1.ShortCut := ShortCut(word('F'), [ssCtrl]);
    SendToBack1.ShortCut := ShortCut(word('B'), [ssCtrl]);
end;

procedure TForm1.FormDeactivate(Sender: TObject);
begin
    New1.ShortCut := 0;
    Open1.ShortCut := 0;
    Save1.ShortCut := 0;
    Print1.ShortCut := 0;
    Exit1.ShortCut := 0;
    Undo1.ShortCut := 0;
    SelectAll1.ShortCut := 0;
    Copy1.ShortCut := 0;
    Cut1.ShortCut := 0;
    Paste1.ShortCut := 0;
    Delete1.ShortCut := 0;
    Group1.ShortCut := 0;
    Ungroup1.ShortCut := 0;
    MakeBlock1.ShortCut := 0;
    BringToFront1.ShortCut := 0;
    SendToBack1.ShortCut := 0;
end;

procedure TForm1.InsertBlock1Click(Sender: TObject);
begin
    iCAD.ToolIdx := toInsertBlock;
end;

procedure TForm1.InsertPicture1Click(Sender: TObject);
begin
    iCAD.ToolIdx := toInsertBitmap;
end;

```

```

procedure TForm1.Combine1Click(Sender: TObject);
begin
  iCAD.FlipImagesOfSelection(fmHorz);
end;

procedure TForm1.Break1Click(Sender: TObject);
begin
  iCAD.FlipImagesOfSelection(fmVert);
end;

procedure TForm1.BringToFront2Click(Sender: TObject);
begin
  iCAD.OrderSelection(osFront);
end;

procedure TForm1.SendToBack2Click(Sender: TObject);
begin
  iCAD.OrderSelection(osBack);
end;

procedure TForm1.BringForward2Click(Sender: TObject);
begin
  iCAD.OrderSelection(osFward);
end;

procedure TForm1.SendBackward1Click(Sender: TObject);
begin
  iCAD.OrderSelection(osBward);
end;

procedure TForm1.ArcStyleNormal1Click(Sender: TObject);
begin
  iCAD.ArrangeArcStyleOfSelection(0);
end;

procedure TForm1.ArcStylePie1Click(Sender: TObject);
begin
  iCAD.ArrangeArcStyleOfSelection(1);
end;

procedure TForm1.ArcStyleChord1Click(Sender: TObject);
begin
  iCAD.ArrangeArcStyleOfSelection(2);
end;

procedure TForm1.ImportDXF1Click(Sender: TObject);
begin
  if DXFDial.Execute then
  begin
    iCAD.ImportDXF(DXFDial.filename,false,false);
  end;
end;

procedure TForm1.iCADSelectionChange(Sender: TObject);
var a,b: integer;
    selFigure: TFigure;
    p1,p2,p3,p4: TPoint;
    ptstr: string;
    p: Tpoint;
begin
  For a:=0 to iCAD.selection.count-1 do
  begin

```

```

SelFigure := TFigure(iCAD.Selection[a]);
if SelFigure is TLine then
begin
  p1 := SelFigure.FigurePoints[1];
  p2 := SelFigure.FigurePoints[2];
  ptstr := 'Points: ';
  for b := 1 to SelFigure.PointCount do
  begin
    p := SelFigure.FigurePoints[b];
    ptstr := ptstr + '['+ inttostr(p.x)+', '+inttostr(p.y)+']';
    if b < SelFigure.PointCount then ptstr := ptstr + '-';
  end;
end;
if SelFigure is TRectangle then
begin
  p1 := SelFigure.FigurePoints[1];
  p2 := SelFigure.FigurePoints[2];
  p3 := SelFigure.FigurePoints[3];
  p4 := SelFigure.FigurePoints[4];
  ptstr := 'Points: ';
  for b := 1 to SelFigure.PointCount do
  begin
    p := SelFigure.FigurePoints[b];
    ptstr := ptstr + '['+ inttostr(p.x)+', '+inttostr(p.y)+']';
    if b < SelFigure.PointCount then ptstr := ptstr + '-';
  end;
end;
end;

end;

end;

procedure TForm1.PrintByTiling1Click(Sender: TObject);
var prh,prw: integer;
begin
  if printdialog1.execute then
  begin
    prh := StrToInt(InputBox('HeightOfPaper', 'Enter the height of paper
excluding the margins', '277'));
    prw := StrToInt(InputBox('WidthOfPaper', 'Enter the width of paper
excluding the margins', '190'));
    iCAD.PrintByTiling('i-CAD-'+extractfilename(filename),prh,prw);
  end;
end;

procedure TForm1.iCADHelp1Click(Sender: TObject);
begin
  iCADHelp.Helpfile := 'icad.chm';
  iCADHelp.HelpType := htHTMLhelp;
  iCADHelp.HelpContent;
end;

procedure TForm1.iCADHelpwin951Click(Sender: TObject);
begin
  iCADHelp.Helpfile := 'icad.hlp';
  iCADHelp.HelpType := htWinhelp;
  iCADHelp.HelpContent;
end;

procedure TForm1.iCADHomepage1Click(Sender: TObject);
begin
  form1.Homepage.click;

```

```
end;  
procedure TForm1.iCADSourceFile1Click(Sender: TObject);  
begin  
    form1.SourceExecute.exec('iCADsource','', '');  
end;  
end.
```



Ek 3: about.pas Delphi 5 dosyası

```

(*****
* User Interface Model for Computer Aided Architectural Design
* Copyright (C) 2002 Emrah Turkyilmaz
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
* -----
* I hope You send me a copy of any important changes/additions You do.
*
* E-mail: e.turkyilmaz@iku.edu.tr
*
*****)

unit about;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, BULabel, BUMailTo;

type
  TfrmAbout = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
    iCADMail: TBUMailTo;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmAbout: TfrmAbout;

implementation

{$R *.DFM}

procedure TfrmAbout.Button1Click(Sender: TObject);
begin
  modalresult := mrOk;
end;

```

end.



Ek 4: i-CAD.iwz Installshield Express dosyası

```

[InstallShield Wizard]
iDate=1
iTime=1
Flag=0
ISX.EXE Size=679936
ISX.EXE Date=11:21:52  8.4.1999
ISX.EXE Ver=2.1.2.62
SETUP.EXE Size=45312
SETUP.EXE Date=9:12:24  23.3.1999
SETUP.EXE Ver=3.0.111.0
SETUP.INS Size=81342
SETUP.INS Date=11:26:40  8.4.1999
SETUP.INS Ver=Not available
_INST16.EX_ Size=81342
_INST16.EX_ Date=11:26:40  8.4.1999
_INST16.EX_ Ver=Not available
_INST32I.EX_ Size=294079
_INST32I.EX_ Date=9:12:22  23.3.1999
_INST32I.EX_ Ver=Not available
ISDEPEND.INI Size=6857
ISDEPEND.INI Date=10:13:00  23.7.1999
ISDEPEND.INI Ver=Not available
SWDEPEND.INI Size=9717
SWDEPEND.INI Date=10:18:30  23.7.1999
SWDEPEND.INI Ver=Not available
ICOMP.EXE Size=119808
ICOMP.EXE Date=9:12:24  23.3.1999
ICOMP.EXE Ver=3.00.062
SPLIT.EXE Size=90624
SPLIT.EXE Date=9:12:24  23.3.1999
SPLIT.EXE Ver=3.00.060
PACKLIST.EXE Size=87552
PACKLIST.EXE Date=9:12:24  23.3.1999
PACKLIST.EXE Ver=3.00.060
Version=2.12
DevTool=Delphi 5
Platform=Win32
DisksBuilt=1
DisksDir=i-CAD\650MB\
TabsVisit=111111111000111111
LangNum=0
LangDir=

```

```

[VisualDesign]
AppName=i-CAD
AppExe=
AppExeFile=
Version=0.1
Company=Yildiz Technical University
Title=d:\Desktop\iCADsource\iCAD.exe
TitleType=0
BackgrndBmp=d:\Desktop\iCADsource\logo_50x50_16Color.bmp
BackgrndAlign=4
Backgrnd=1
BackgrndColor=0
Uninstall=1
UninstallChecked=1
Silent=1
SmsMode=0

```

```

[Types]
Types=1
Type1Components=1
Type1Name=Complete
Type1ComponentList=1

[Components]
Components=1
Component1Groups=1
Component1Name=Application Files
Component1Description=Application files needed to run the application
Component1GroupList=1

[Groups]
Groups=1
Group1Size=1556187
Group1Files=5
Group1Name=Program Files
Group1Dir=<INSTALLDIR>
Group1Update=0
Group1TargetOS=-1
Group1File1=d:\Desktop\iCADsource\iCAD.exe
Group1File2=d:\Desktop\iCADsource\iCADsource.exe
Group1File3=d:\Desktop\iCADsource\icad.chm
Group1File4=d:\Desktop\iCADsource\isOzeti.txt
Group1File5=d:\Desktop\iCADsource\license.txt

[Sequence]
DestinationLocationDir=<ProgramFilesDir>\i-CAD
DatabaseLocationDir=<INSTALLDIR>\data
SelectProgramFolderName=i-CAD
SplashScreen=1
SplashScreenBmp=d:\Desktop\iCADsource\logo_50x50 copy.bmp
WelcomeMessage=1
WelcomeMessageFile=
LicenseAgreement=1
LicenseAgreementFile=d:\Desktop\iCADsource\license.txt
ReadMeFileBrowser=1
ReadMeFileBrowserFile=d:\Desktop\iCADsource\isOzeti.txt
DestinationLocation=1
SelectProgramFolder=1
BeginFileTransfer=1
ProgressBar=1
SetupComplete=1
SetupCompleteExec=[Program Files]\iCAD.exe
SetupCompleteExecParam=
SetupCompleteExecReadme=[Program Files]\isOzeti.txt

[RegEntries]
Reg1Path=HKEY_CLASSES_ROOT
Reg1Val1Type=0
Reg1Val1Name=(Default)
Reg1Val1Data=(value not set)
Reg1Vals=1
Reg2Path=HKEY_CURRENT_USER
Reg2Val1Type=0
Reg2Val1Name=(Default)
Reg2Val1Data=(value not set)
Reg2Vals=1
Reg3Path=HKEY_LOCAL_MACHINE
Reg3Val1Type=0
Reg3Val1Name=(Default)

```

```
Reg3Val1Data=(value not set)
Reg3Vals=1
Reg4Path=HKEY_USERS
Reg4Val1Type=0
Reg4Val1Name=(Default)
Reg4Val1Data=(value not set)
Reg4Vals=1
Reg5Path=HKEY_CURRENT_CONFIG
Reg5Val1Type=0
Reg5Val1Name=(Default)
Reg5Val1Data=(value not set)
Reg5Vals=1
Reg6Path=HKEY_DYN_DATA
Reg6Val1Type=0
Reg6Val1Name=(Default)
Reg6Val1Data=(value not set)
Reg6Vals=1
Regs=6
```

```
[Registry]
```

```
[Icons]
Icons=1
Icon1Param=
Icon1Cmd=[Program Files]\iCAD.exe
Icon1Description=iCAD
Icon1WorkingDir=[Program Files]
Icon1IconFile=
Icon1RealFile=d:\Desktop\iCADsource\iCAD.exe
Icon1WhichIcon=0
Icon1KeyVirtual=0
Icon1KeyFlags=0
Icon1StartMenu=0
Icon1Window=0
```

```
[Autoexec]
PathDir=
ShareLocks=
ShareFiles=
Command=
```

```
[CopyToFloppy]
DriveOption=Path
DestinationPath=D:\
DestinationPathOneFile=
Password=
```

```
[Uninstall]
UserUninstall=0
```

Ek 5: i-CAD CD'si (i-CAD "kur" d... i, yardım... i, program kodları)



ÖZGEÇMİŞ

Doğum tarihi 01. 10. 1976

Doğum yeri Kırklareli

Lise 1989-1994 Kadıköy Anadolu Lisesi

Lisans 1994-1999 Yıldız Teknik Üniversitesi Mimarlık Fakültesi
Mimarlık Bölümü

Yüksek Lisans 1999-2002 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Ortamında Mimarlık Programı

Çalıştığı kurumlar

1999-2000 Baran-Mer International İstanbul Liaison Office
2000-2001 İdeYapı LTD. Şti.
2001- İstanbul Kültür Üniversitesi

