

YILDIZ TEKNİK ÜNİVERSİTESİ FEN
BİLİMLERİ ENSTİTÜSÜ

SAYISAL TASARIM KAVRAMLARI ve ALGORİTMİK
DÜŞÜNCENİN MİMARİ TASARIMA ETKİLERİ

Mimar Alp ÇILTIK

FBE Mimarlık Anabilim Dalı Bilgisayar Ortamında Mimarlık Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Oya PAKDİL (YTÜ)

İSTANBUL, 2008

SAYISAL TASARIM KAVRAMLARI VE ALGORİTMİK DÜŞÜNCENİN MİMARİ TASARIMA ETKİLERİ

ÖZET

Bilgisayarlar ve oluşturdıkları sistematik duruş, her disiplinde olduğu gibi mimarlık disiplini de etkilemektedir. Son yıllarda bu sistematik duruş, mimari tasarımın kuramsal alt yapısını ve söylemlerini değiştirmektedir. Mimari tasarım süreci rasyonel bir süreç olarak görülmemektedir. Tasarım probleminin matematiksel verileri ve tasarımcının altyapısı ile ortaya konan çift katmanlı süreçtir. Bu katmanlar tasarım sürecinde kesişir, evrimleşir, hiyerarşik olmayan biçimlerde dallanır ve yeni katmanlar oluşturabilir.

Tasarım süreci kompleks fenomeni içinde değerlendirilmektedir. Kompleks sistemler ve kendini organize eden sistemleri çözümlmek için soyut makineler araç olarak kullanılmaktadır. Soyut makineler ya da başka deyişle algoritmalar ile tasarım sürecini organize etmek yeni bir fikir değildir. Günümüzde bu fikrin üzerinde tekrar yoğunlaşmaktadır. Bilgisayar destekli temsili olağan bir hal aldığı mimarlık disiplini, paket programları kendi isteğine göre değiştirerek ve yazılımlara çeşitli eklemeler (*script*) yaparak bilgisayar destekli tasarımın sınırlarını aramaktadır. Bu arayışın fikir altyapısı, Deleuze'un ortaya koyduğu kavramlar ve farklı disiplinlerin (biyoloji, matematik, termodinamik, felsefe, dil bilimi, bilgisayar) araştırma konusu olan evrim, genetik, topoloji, algoritma, kendini organize eden sistemler, kompleks sistemler, dil ve kod sistemleri gibi kavramlarla desteklenmektedir. Tüm bu kavramlar ile mimarlık kavramları arasında analogiler kurulmaktadır. Sonuçta tasarım süreci ve "algoritmik düşünce biçimi" çalıştırılmaktadır.

Anahtar kelimeler: Tasarım, Mimari tasarım, Bilgisayar destekli tasarım, Algoritmik düşünce, Soyut makineler, Algoritmik tasarım, Parametrik tasarım, Hesaplamalı tasarım, İşleme

DIGITAL DESIGN CONCEPTS, ALGORITHMIC THINKING AND EFFECTS ON ARCHITECTURAL DESIGN

ABSTRACT

Computers, and the systematical posture they formed, affect all the disciplines as well as architecture. In the recent years, this systematical posture changes the theoretical substructure and the motto of architectural design. The process of architectural design is not seen as a rational process. Two-layered process which displays both the mathematical data of the design problem and substructure of the designer. In the design process these layers intersect, evolves, branch into non-hierarchical forms and develop new layers. The design process is being evaluated in the 'complexity' phenomenon.

Abstract machines are used as tools to analyze complex systems and self-organized systems. To organize design process with abstract machines –or in other words algorithms – is not a new idea. Nowadays it is being majored on this idea again. The architectural discipline, in which computer aided representation becomes an ordinary procedure, is seeking the limits of computer aided design by manipulating package programs or by adding scripts to the software. The substructure idea of this search is based on the concepts stated by Deleuze and the concept of research subject of different disciplines (biology, mathematics, thermodynamic, philosophy, linguistics and computer) such as evolution, genetics, topology, algorithm, self-organized systems, complex systems, language, and code systems. All these concepts and architectural concepts are combined analogically. Finally, design process and the way of algorithmic thinking could be juxtaposition.

Keywords: Design, Architectural design, Computer aided design, Algorithmic thinking, Abstract machines, Algorithmic design, Parametric design, Computation.

İÇİNDEKİLER

	Sayfa
KISALTMA LİSTESİ	iii
ŞEKİL LİSTESİ	iv
ÖNSÖZ.....	v
ÖZET	vi
ABSTRACT	vii
1. GİRİŞ	1
1.1 Amaç	3
1.2 Kapsam	4
1.3 Yöntem	4
2. DİJİTAL TASARIM KAVRAMLARI VE DELEUZE	6
2.1 Popülasyon Düşüncesi ve Çokluk Kavramı	8
2.1.1 Tipolojik Bakış	8
2.1.2 Popülasyon Düşüncesi	8
2.2 Özgül (İçlem) Düşünce.....	11
2.3 Topolojik Düşünce.....	12
3. MİMARİ TASARIM VE ALGORİTMİK DÜŞÜNCE	17
3.1 Algoritma ve Algoritmik Düşünce nedir?	17
3.1.1 Kompleks Sistemler, Kendini Organize Eden Sistemler, Kolektif Zeka.....	22
3.1.2 Evrensel, Soyut makineler ve İşlemeleme (Computation)	24
3.2 Tasarım Nedir?	28
3.3 Tasarım ve Algoritma	32
3.3.1 Tasarımcı ile Bilgisayar Kodlaması.....	34
3.3.2 Algoritmik Mimari Tasarım	37
4. TASARIMIN ALGORİTMİK DÜŞÜNCE İLE DENEYLENMESİ.....	42
4.1 Deneye Hazırlık	43
4.1.1 Tasarım Probleminin Analitik ve Algoritmik Tabana Aktarılması	44
4.2 Deney Süreci.....	51
4.3 Deney Sonucu.....	55
5. DEĞERLENDİRME VE SONUÇLAR	58
KAYNAKLAR.....	61
EKLER	65
ÖZGEÇMİŞ.....	86

KISALTMA LİSTESİ

NURBS	Non-Uniform Rational B-Splines
BDMÇ	Bilgisayar Destekli Mimari Çizim
BDMT	Bilgisayar Destekli Mimari Tasarım
BDT	Bilgisayar Desteli Tasarım
BDÜ	Bilgisayar Destekli Üretim

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Tip ve popülasyon düşüncesi arasındaki fark diyagramı.	9
Şekil 2.2 Mobius Bantı	13
Şekil 3.1 Abaküs.....	18
Şekil 3.2 Jacquard Tezgahı	19
Şekil 3.3 Delikli kart.....	20
Şekil 3.4 Akış Diyagramı hazırlarken kullanılabilecek şekiller	20
Şekil 3.5 Örnek Akış Diyagramı	21
Şekil 3.6 Termit kuleleri.....	23
Şekil 3.7 Sihirbaz Tasarımcı - Bilgisayar Tasarımcı.....	29
Şekil 3.8 Örümcek Ağı	30
Şekil 4.1 İlk Eskiz.....	44
Şekil 4.2 İlk eskizin analitik soyutlaması	45
Şekil 4.3 Fikrin analitik ve algoritmik soyutlanması.....	45
Şekil 4.4 Yazılan algoritmanın akış diyagramı	46
Şekil 4.5 Script arayüzü.....	47
Şekil 4.6 Grid kesişimlerindeki vertex noktaları	48
Şekil 4.7 Manipülasyon değerleri.....	48
Şekil 4.8 Script kullanımı sırasında odak noktaları belirlenmesi	49
Şekil 4.9 Script kullanımı sırasında diğer manipülatörlerin belirlenmesi	50
Şekil 4.10 Deney öncesi ortaya çıkabilecek ürün araştırmasından örnekler	51
Şekil 4.11 Atölye çalışmasından görünüm	52
Şekil 4.12 Atölye çalışmasından görünüm	52
Şekil 4.13 Atölye çalışmasından eskiz aşaması ve üç boyutlu ürün üretimi aşaması	53
Şekil 4.14 Atölye çalışmasından üç boyutlu ürün üretimi	54
Şekil 4.15 Atölye çalışmasından üç boyutlu ürün üretimi	55
Şekil 4.16 Bir kullanıcının deney sürecinin adımları (EK 2)	56

ÖNSÖZ

Bana bu tez konusunu araştırmaya yönlendiren merakın altyapısını verdikleri için Yıldız Teknik Üniversitesi, Bilgisayar Ortamında Mimarlık Kürsüsü öğretim görevlilerine ve tez çalışması sürecine gerekli zamanı ayırmama destek olan işverenim Hüseyin Yüzbaşıoğlu'na teşekkür ederim. Tezimi yazarken, araştırmalarım sırasında ve atölye çalışmalarında yanımda olan arkadaşlarıma yardımlarından dolayı çok teşekkür ederim.

Tez danışmanım Prof. Dr. Oya Pakdil'e bana olan güveni ve desteği için özellikle teşekkür ederim.

Alp ÇILTIK

Mehmet Bölük'ün anısına...

SAYISAL TASARIM KAVRAMLARI VE ALGORİTMİK DÜŞÜNCENİN MİMARİ TASARIMA ETKİLERİ

ÖZET

Bilgisayarlar ve oluşturdıkları sistematik duruş, her disiplinde olduğu gibi mimarlık disiplini de etkilemektedir. Son yıllarda bu sistematik duruş, mimari tasarımın kuramsal alt yapısını ve söylemlerini değiştirmektedir. Mimari tasarım süreci rasyonel bir süreç olarak görülmemektedir. Tasarım probleminin matematiksel verileri ve tasarımcının altyapısı ile ortaya konan çift katmanlı süreçtir. Bu katmanlar tasarım sürecinde kesişir, evrimleşir, hiyerarşik olmayan biçimlerde dallanır ve yeni katmanlar oluşturabilir.

Tasarım süreci kompleks fenomeni içinde değerlendirilmektedir. Kompleks sistemler ve kendini organize eden sistemleri çözümlmek için soyut makineler araç olarak kullanılmaktadır. Soyut makineler ya da başka deyişle algoritmalar ile tasarım sürecini organize etmek yeni bir fikir değildir. Günümüzde bu fikrin üzerinde tekrar yoğunlaşmaktadır. Bilgisayar destekli temsili olağan bir hal aldığı mimarlık disiplini, paket programları kendi isteğine göre değiştirerek ve yazılımlara çeşitli eklemeler (*script*) yaparak bilgisayar destekli tasarımın sınırlarını aramaktadır. Bu arayışın fikir altyapısı, Deleuze'un ortaya koyduğu kavramlar ve farklı disiplinlerin (biyoloji, matematik, termodinamik, felsefe, dil bilimi, bilgisayar) araştırma konusu olan evrim, genetik, topoloji, algoritma, kendini organize eden sistemler, kompleks sistemler, dil ve kod sistemleri gibi kavramlarla desteklenmektedir. Tüm bu kavramlar ile mimarlık kavramları arasında analogiler kurulmaktadır. Sonuçta tasarım süreci ve "algoritmik düşünce biçimi" karşılaştırılmaktadır.

Anahtar kelimeler: Tasarım, Mimari tasarım, Bilgisayar destekli tasarım, Algoritmik düşünce, Soyut makineler, Algoritmik tasarım, Parametrik tasarım, Hesaplamalı tasarım, İşleme

DIGITAL DESIGN CONCEPTS, ALGORITHMIC THINKING AND EFFECTS ON ARCHITECTURAL DESIGN

ABSTRACT

Computers, and the systematical posture they formed, affect all the disciplines as well as architecture. In the recent years, this systematical posture changes the theoretical substructure and the motto of architectural design. The process of architectural design is not seen as a rational process. Two-layered process which displays both the mathematical data of the design problem and substructure of the designer. In the design process these layers intersect, evolves, branch into non-hierarchical forms and develop new layers. The design process is being evaluated in the 'complexity' phenomenon.

Abstract machines are used as tools to analyze complex systems and self-organized systems. To organize design process with abstract machines –or in other words algorithms – is not a new idea. Nowadays it is being majored on this idea again. The architectural discipline, in which computer aided representation becomes an ordinary procedure, is seeking the limits of computer aided design by manipulating package programs or by adding scripts to the software. The substructure idea of this search is based on the concepts stated by Deleuze and the concept of research subject of different disciplines (biology, mathematics, thermodynamic, philosophy, linguistics and computer) such as evolution, genetics, topology, algorithm, self-organized systems, complex systems, language, and code systems. All these concepts and architectural concepts are combined analogically. Finally, design process and the way of algorithmic thinking could be juxtaposition.

Keywords: Design, Architectural design, Computer aided design, Algorithmic thinking, Abstract machines, Algorithmic design, Parametric design, Computation.

1. GİRİŞ

Bilgisayar teknolojilerinin gelişmesi, tüm disiplinleri ve alanlarını etkilediği gibi, mimarlık kavramını ve alanını da etkilemektedir. Günümüzde mimarlar, bilgisayar destekli mimari çizim (BDMÇ) ürünlerinden en az birini kullanmaktadır. Bu ürünler, mimari sunuma belirli bir standart getiren teknik çizim, modelleme ve animasyon üretmek amaçlı kullanılmaktadır. Mimarlık alanında bilgisayar desteği üzerine araştırmalar 60'lı yıllarda başlamıştır. Buna karşın pratik kullanımı olmamıştır. Her alanda olduğu gibi, mimarlıkta da bilgisayarın pratik alanda kullanılması, “*Kişisel Bilgisayar*”ların (PC – Personal Computer) 80'lerin ortasında üretilmesiyle başlamıştır. Bilgisayarlar 80'li yıllarda ilk olarak çizim aracı olarak kullanılmıştır -Autodesk, Graphisoft vs.-. Ancak uzun süre mimarlar geleneksel çizim yöntemlerine (kağıt ve kalem) güvenmeye devam etmiştir. 90'ların başında BDMÇ aracı olarak kendini kabul ettirmeye başlayan bilgisayar, bundan böyle mimarlık disiplini açısından daha ön plana çıkmaya başlamış ve günün teknoloji limitlerini zorlamaya başlamıştır.

Bilgisayarın mimarlık sektörüne uyumu sürecinin başlangıcında, *üç boyutlu doğrusal olmayan çizgilerin* (NURBS - Non-Uniform Rational B-Splines) kullanımı, 90'ların ortasında popüler olmakta ve daha önce üzerinde çalışılmayan geometriler üzerine gidilerek bu geometrilerin, tasarım ve üretim bandında kullanımının artmasına sebep olmaktadır. Daha karmaşık geometrili bir tasarımın ürün olabilmesi için gerekli üretim biçimi de, hava ve su dinamiğinin önemli olduğu gemi ve uçak sanayisi için geliştirilen CNC (Computer Numerical Control) makinelerini kullanılarak oluşturulmaktadır. Bu üretim bandının kullanımı öncesinde, mimari ürünlerde kendine çok fazla yer bulamayan çok eğrilikli yüzeylerin yaygın biçimde kullanılabilmesini sağlamıştır. Mimarlık kavramı, sadece üretim açısından değil diğer alanlarında da gözle görülür bir değişim içerisindedir.

Temsil, sunum ve üretim sürecinde bilgisayarların ağırlığı hissedilmektedir. BDMÇ'nin mimari ürün üretme sürecine entegrasyonu, üretimin sayısal kesim CNC makinelerinin de devreye girmesiyle bu sürecin akışına önemli etkileri olmuştur.

Sayısal mimarlık, kendi tarihi süreci içerisinde, önceleri biçim üretme kaygısına dayalı olarak çok eğrilikli yüzeyler kullanılması ile özdeşleştirilmekteydi. Bu mimarlık alanında daha önce kullanılmayan biçimlerin kullanılması sebebiyle başarı olarak görülmekteydi. Zamanla mimarlığın biçimsel kaygıları dışında kalan diğer alanların da üzerinde durulmaya başlanmaktadır. Bilgisayarlar, tasarım sürecinde, sürecin akışına etki edecek kadar etkili bir

araç olmaya başlamıştır. Günümüzde sayısal mimarlık biçimsel kaygının ötesine geçerek tasarım sürecini irdelemektedir. “BDT (Bilgisayar Destekli Tasarım) ve BDÜ (Bilgisayar Destekli Üretim) teknolojileri, bir araya getirmesi günümüze kadar çok zor ve pahalı olan, karmaşık mimari strüktürleri, ürün olarak ortaya koyarak, mimarların önüne yeni tasarım alanları sunmaktadır. Fakat “sayısal araçlar” tasarımcıya düşünme sürecinin çok başlangıcında destek olabilmektedir.” (Kolarevic, 2001). Bilgisayarı mimari sunum ve temsil aracı olması dışında yeni bir tasarım ortamı olarak değerlendiren mimarlar, bu konuya henüz deneysel olarak yaklaşmaktadırlar. Tasarımın ilk düşünme sürecinden üretimin bitimine kadar süreklilik sağlanmaya çalışılmaktadır.

Günümüzde, *mimar-bilgisayar iletişimi*yle ilgilenen araştırmacı mimarlar, tasarım sürecine yeni bir bakış ile yaklaşmaktadırlar. Bilgisayar kullanımının tasarım sürecinin içine entegrasyonu üzerine çalışan disiplinler ise bilgisayarın tasarım süreci içindeki potansiyellerini araştırmaktadırlar.

Bilgisayarlar son dönemde *sosyo-kültürel akışı* değiştirmiştir ve kendi içlerindeki çalışma prensipleri, birbirleri arasında kurdukları bağlar, internet ve internetin işleyiş dinamikleri toplum bilimin bir konusu haline gelmiştir. Sosyal olarak kurulan dolaylı bağlar, internetin kendi içerisinde bir *kolektif zeka* olarak işlemesini sağlamaktadır.

Dijital tasarım kavramlarının algılanması için bilgisayarın çalışma mantığını algılamak gerekmektedir. Bu mantığı çözmek, algoritmaları anlamak ile başlamaktadır. Bir problemin ideal çözümüne giden yola *algoritma* denir. Bir başka deyişle; algoritma bir probleme yaklaşım biçimidir. Algoritma bir problemin çözüm yollarının adımlanmasıdır. Programın algoritması, programlama dilinden daha önemlidir. Programı çalıştıracak algoritmayı en iyi şekilde çözümledikten sonra, kullanılacak dilin yapısına göre kodlama aşamasına geçilir. Bu noktada, kullanılan kodlama bir bilgisayar dili değil *mimari kodlama* biçimi (diğer deyişle mimari teknik dili) olabilir. Bilgisayar mantığının algoritmalar üzerinde çalışmasının sebebi bilgisayarın sadece *matematiksel mantıkla* çalışmasıdır. Bilgisayarın, tasarımın sürecine entegre bir araç olabilmesi için, tasarım sürecinin tasarımcı tarafından tasarlanması gerekmektedir. Bu açıdan bakıldığında tasarımcının bir tasarı probleminin çözümüne yaklaşımı, *algoritmik ve matematiksel mantıkla* tanımlanabilir olması gerekmektedir. 21. yüzyılın mimarlara getirdiği yeni araştırma alanlardan biri de “*tasarımın sayısallaştırılması*” olmaktadır.

“Tasarım çabasının, kendine has bir alanı olan *sayısal tasarım*; farklı içeriği disiplinler arası teorik kaynaklarıyla motive edilerek, kendine ait söylemi ile ortaya çıkarılarak ve yeni teknolojiler ile desteklenerek tasarımın özgün sınıfı olarak *evrimini* ortaya koymakta ve son on yılda hızla kristalize olmuş bir olgu haline gelmektedir.” (Oxman, 2005) Bilgisayara *akıllı tasarım desteği* olarak bakıldığında, bilgisayar destekli tasarım veya bilgisayar destekli mimari tasarım, tasarım fikir ve teorilerinin de tekrar gözden geçirilmesi gerektiğini ortaya koymaktadır.

Günümüzde tüm disiplinler sayısallaşan bilgi akışından etkilenmektedir. Bilgi akışının hızının artması mimari ürünün yapım sürecinde mimarlara esneklikler sağlamıştır. Mimarlar ürünlerini başka kıtalarda tasarlayıp başka kıtalarda uygulayabilmektedir. Mimari ürünün ihtiyacı olan mühendis desteği de diğer bir üçüncü kıtadan alınabilmektedir. Bilgi akışı sayısal ortamda internet ve bilgisayarlar aracılığıyla sağlanmaktadır. Bu da ürünün sayısal ortamlarda temsiliyle olmaktadır. (Kolarevic, 2001)

Günümüzde “fikirden ürüne” *süreklilik* kavramı söz konusudur. Hatasız ürün üretmek ve “*fikrin üründe daha olgun belirmesi*” amaçlanmaktadır. Mimarlık disiplini, tasarım sürecinin *dolaylı* değil birebir etkilendiği bir dönüşüm geçirmektedir.

1.1 Amaç

Tasarımın erken süreci, çözümlenemeyen ve değişebilen girdileri sebebiyle, derinlemesine araştırmanın dışında tutulmaktadır. Bu sebeple, tasarım süreci üzerine yapılan araştırmalar yeterli olgunluğa ulaşmaktan uzak kalmıştır. Bunun yerine tasarım sonrası adımlar üzerine uzmanlaşmış ve bunun üzerine çalışılmıştır. Günümüzde ise araştırmalar, bilgisayarın hesaplama gücü, depolama imkanları ve organize edebilme yetenekleri, tasarımın erken aşamalarında tasarımcının, dijital teknikler tarafından yararlı bir biçimde yönlendirebileceğine dair somut örnekler verebilmektedir.

Sayısal tasarım, teorik çatısını günden güne ortaya koymaktadır. Sayısal tasarım söylemleri, uygulamadaki değişimlerle eskisinden daha güçlü motive edilmektedir. Dijital tasarım günümüzde pratikteki araştırmaları, teori üretimleri ve bazı ürün verme denemeleri sürecinden geçmektedir. Bu tezde algoritmik tasarımın kavram çatısı ve teorik modelinin analiz edilmesi çabası olacaktır. Bu analiz bazında amaçlanan *fikir*, *bilim* ve *mimari* kavram ve kuramların günümüz mimarlık bakışına verdiği yönü algılamak olacaktır.

Genel olarak tezin amacı, dijital tasarım kavramlarını, algoritmik düşünce fikri çerçevesinde toparlamak ve algoritmik tasarımın incelenmesi olacaktır. Dijital tasarım sürecinde tasarımcılar, paket programları kendi isteğine göre değiştirerek ve yazılımlara çeşitli eklemeler (*script*) yaparak bilgisayar destekli tasarımın sınırlarını aramaktadır. Bu yazılım eklemelerinin ya da başka bir deyişle *script* yazımının, tasarım sürecine katılabilmesindeki belirsizlikler bağlamında tez içerisinde yapılan özgün çalışmanın alana katkı sağlaması amaçlanmaktadır.

1.2 Kapsam

Araştırma kapsamı, algoritmik mimari tasarım üzerinedir. Tez kapsamında dijital tasarım özellikle kuramsal alt yapısı irdelenecek, dijital tasarım söyleminin kavramlarının açılımı yapılacaktır. Farklı disiplinlerin araştırma konusu olan kavramlarla, mimarlık kavramları arasında analogiler kurularak tasarım sürecinin değişimi anlaşılmasına çalışılacaktır. Tez kapsamında özellikle algoritmik düşünce biçiminin, anlaşılması için bu düşünce biçiminin arkasındaki fikir ve bilim kavramları irdelenecektir. Mimari tasarım sürecinde son kullanıcı tarafından yazılım üretilmesi için kullanımı, üzerine durulacaktır.

İlk bölümde konunun kısa tarihi ve mimari tasarımın bulunduğu eşik işaret edilecektir. İkinci bölümde konunun fikir altyapısı, Deleuze ve onun çeşitli bilim dallarından aldığı kavramlar, açıklanacaktır. Üçüncü bölümde ise, kompleks kavramıyla değerlendirilen tasarım sürecinin algoritmik çözümü üzerine durulacaktır. Tasarım ve algoritmik düşünce kavramları üst üste çakıştırılacaktır. Dördüncü bölümde ise, bu örtüşmeye örnek teşkil edebilecek bir atölye çalışması anlatılacak ve ortaya çıkan verileri değerlendirme üzerinde durulacaktır. Son bölümde ise bu tartışılan kavramlar mimari kavramlar deneysel çalışmanın ortaya koyduğu veriler genel bir çerçevede değerlendirilecek ve sonuçlar üretilmeye çalışılacaktır.

1.3 Yöntem

Araştırma süresince;

Günümüzde dijital tasarımın kuramsal çerçevesinin dayandırıldığı ana kuram ve fikirlerin üzerine gidilerek bu kavram ve fikirler üzerine literatür araştırması yapılmıştır.

Tasarım sürecinin değişimini algılamak amaçlı tasarım sürecinin çözümlenmesi için disiplinler arası literatür taramaları yapılmıştır, mimarlık ve diğer disiplinlerin (mühendislik,

biyoloji, matematik, programlama, felsefe vs.) ortak ürettiği kavramlar ile dijital tasarım adına verilmiş örnekler üzerine gidilerek konunun irdelenmesi yapılmıştır. Tasarımcı ve bilgisayar arası alışverişin yönelimi gözlenmiştir. Tasarım sürecini, kompleks olgusu ile örtüştürme çabasıyla evrim kavramı, kendini organize eden sistemler ve kompleks sistemler üzerine gidilmiştir. Bu sistemleri matematiksel olarak çözümlemeye yönelik çalışmalar olan hücresel otomat, evrensel soyut makineler gibi araçlarla birlikte algoritmik düşünce ve işleme (computation) kavramı irdelenmiştir. Algoritmik düşünce biçimi kavramlarının yerleri, felsefe, bilim ve mimarlık disiplinlerinin oluşturduğu üçgen içerisindeki araştırılmıştır.

- Dijital tasarım üzerine yapılan kavramsal çalışmaların incelenmesi gerekmiştir.
- Dijital tasarım kavramları ile tasarım süreci ve tasarımcı arasındaki teorik ilişki incelenmiştir.
- Tasarım süresinde dijital teknolojilerin özellikle tasarımın başlangıç aşamasına etki eden kavramlar araştırılarak incelenmiştir.
- Tasarım sürecinin gelecekte alabileceği biçim yorumlanmıştır.
- Autodesk firmasına ait 3D Studio MAX programının içeriğinde bulunan MAXScript ile bir tasarım fikri analitik ortama ve ardından algoritmik olarak sayısal ortama aktarılmıştır.
- Oluşturulan yazılım Bilgisayar Ortamında Mimarlık Kürsüsü Laboratuvarında bir mimar grubuyla denenmiştir.
- Bu gruptan fikirlerini bir anket yoluyla belirtmeleri istenmiştir.
- Grubun fikirleriyle birlikte araştırmanın ana kapsamı örtüştürülmüştür.

2. DİJİTAL TASARIM KAVRAMLARI VE DELEUZE

Mimari ürün üretme süreci; tasarım süreci, temsil süreci ve üretim süreci olarak üç aşamalı bir süreç olarak değerlendirilebilir. Bilgisayar, göreceli olarak kısa denebilecek bir sürede bu üç aşamalı sürecin son iki dönemin yoğun olarak kullanılır hale gelmiştir. Mimari projelerin uygulama aşamasında, Saarinen'in TWA terminalinde olduğu gibi topolojik yüzeylerle tekrar ilgilenilmeye başlanmıştır. BDÜ öncesinde, üretimi hem taşıyıcılık hesaplamalarındaki zorluklardan hem de kalıpların veya kesimlerin, proje uygun yapılmasındaki zorluklardan dolayı topolojik geometrilerden kaçınılmaktaydı. BDÜ sayesinde, projelerde çok eğrili **topolojik geometrilerin** kullanımı artmaktadır. Üretimdeki gelişmelerin yanı sıra mimari ürün temsili de bilgisayar üzerinden yapılmaktadır. Bu sebeple mimari tasarım ve mimari ürün temsili ilişkisi sorgulanır hale gelmiştir. Bu gelişmeler bağlamında mimari ürün üretme sürecinin aşamalı bir süreç olmadığı söylemleri savunulmaktadır. Süreç bir bütün ve **sürekli**dir. Tasarım fikir olarak doğduğu andan itibaren ürünün üretimine kadar **süreklilik** içinde olmalıdır. Bu sebeple bilgisayarın erken tasarım sürecine entegrasyonu tartışılmaktadır. Bu tartışma sürecinde mimarlar, disiplini bu noktaya getiren fikir altyapısını tekrar değerlendirmektedir. Öncelikle dijital tasarım kavramları ardından bilgisayar bilimleri kavramlarını incelemektedir. Bu sırada birbirinin içine girmiş bu kavram bulutu içerisinde özellikle biyoloji ve matematiğin yerlerini netleştirmeye çalışılmaktadır. Tasarım süreci tekrar değerlendirilmektedir. Bilgisayar ile tasarım süreci iletişiminin potansiyelleri araştırılmaktadır. Bilgisayarın algoritmik sistemle çalışması sebebiyle tasarım süreci mümkün olduğunca şeffaflaştırılmaya çalışılmaktadır.

Mimarlık disiplini, felsefî ve teorik olarak çeşitli kaynaklardan beslenmektedir. Sayısal mimarlık, mimarlığa bakışıyla “*Bauhaus*”, organik ve biyomorfik geometrik biçimleri kullanması sebebiyle “*Neo-Barok*”, “*Art-Nouveau*” hatta “*Sürrealist ekspresyonist*” olarak değerlendirilebilmektedir. Ütopik bir geleceğin mimari dili olabileceği savıyla “*archigram*” ile ilişkisi de kurulabilmektedir. Tüm bu açılımlara rağmen, sayısal mimarlık, temel kavramlarını Alman filozof ve matematikçisi Gottfried Wilhelm Leibniz (1646-1716) ve 20. yüzyılın en etkili düşünce adamı Gilles Deleuze (1925 – 1995) ün fikir ve teorilerinden almaktadır. Doğrusal olmayan (Non-Linearity), bağlar arası (inter-connectivity), süreklilik (continuity), iletişim ağı (networks), dinamizm (dynamism), şema (diagram), tasarım makineleri (design machines), soyut makineler (abstract machines), hiper-bağlar (hyper-

connectivity), Hiyerarşik olmayan strüktürler (non-hierarchical structures) Deleuze fikir ve teorilerinin içinde var olan kavramlardır. (Oxman 2005)

Günümüzde mimarlık sözü edilen kavramlardan etkilenmektedir. DeLanda sayısal tasarım söyleminde Deleuze’ün önemi için, “Tasarım çabası, zengin üç boyutlu uzayında kesin felsefi fikirler arar. Bu yolda yol gösterici olarak Gilles Deleuze önemi bir rol oynamaktadır. Algoritmaların yaratıcı kullanımı, üç felsefi düşünce biçimine yayılarak çözümlenebilmektedir. Popülasyon, özgül (işlem) ve topolojik düşünce (populational, intensive, topological thinking) biçimleri Deleuze tarafından keşfedilmiş değildir. Fakat Deleuze bunları bir araya getirerek yepyeni bir biçim üretim konseptine temel olmaktadır” demektedir. (DeLanda 2001)

Bu kavramlar dijital tasarım fikrinin zeminini örmektedir. Bu zemin üzerine, yine bilimsel ve mimari kavramlar, birbiriyle hiyerarşik ve doğrusal olmayan, dinamik, açık uçlu bir örüntü oluşturmaktadır. Bu bölümde fikir altyapısını anlamak için sırasıyla “*Popülasyon Düşüncesi*”, “*Özgül Düşünce*” ve “*Topolojik Düşünce*” kavramları açıklanacaktır. Deleuze’ün varlık felsefesinin önemli kavramlarından olan “*Çokluk*” üzerinde durulmaktadır.

“Popülasyon Düşüncesi” varlıklar arasındaki benzerlikleri vurgulayarak tipleşmek yerine, *farklılıkları* vurgulayarak *bireyselleşmek* gerektiğini söylemektedir. Farkın üretken olduğunu belirtmektedir. “Özgül düşünce” ise varlığın, farklılıkları üzerinden bireyselleşirken, *ana karakteristiklerini kaybetmemesi* gerektiği ya da varlığın bütününde bir *öz* bulunması gerektiğini söylemektedir. “Topolojik düşünce” ise bu *bireysel öz*’ün esnek, dönüştürülebilir, sürdürülebilir olması gerektiğini vurgulamaktadır. Ayrıca bütünü oluşturan parçalarının birbiriyle olan ilişkilerinin, topolojik bir biçimde kurulacağını söylemektedir. Topolojik geometriler, farklar geometrisi olmaktadır.

Sayısal teknolojiler, mimari tasarımı, yeni kavramlar ile tanıştırmaktadır. Sayısal Tasarım, sayısal mimarlık, sayısal düşünce, Öklid dışı geometrik uzay, topolojik düşünce biçimi, kinetik ve dinamik tasarım, parametrik tasarım, türetici tasarım, genetik algoritmalar gibi yeni kavramlar mimarlık kavram dünyasında yerini almaktadır. (Kolarevic, 2001)

Dijital mimarlık ya da sayısal mimarlık yüzyılımızın başında bu ana fikir üzerinden biçimlenmeye ve türemeye çalışmaktadır.

2.1 Popölasyon Dölşölncesi ve Çokluk Kavramı

Popölasyon dölşölncesi, Deleuze tarafından biyolojiden analogi yoluyla alınmıřtır. Popölasyon dölşölncesi kavramını ilk önce 20. yüzyılda evrim teorisyenin modern bir sentezini yapan Ernst Walter Mayr (1904, Almanya – 2005 A.B.D) tarafından kullanılmıřtır. [5] Bu kavramı anlayabilmek için öncelikle yine Mayr'ın ortaya koyduđu diđer bir kavram olan “Tipolojik Dölşölnce”yi incelemek gerekmektedir.

Biyoloji felsefesinin popüler olan iki kavramı Tipolojik Dölşölnce ve Popölasyon Dölşölncesi üzerine yoğun tartışmalar antik çağdan günümüze kadar süregelmıřtir.[3]

2.1.1 Tipolojik Bakıř

Tipolojik Dölşölnce, tözcölük (Essentialism) akımının bir yansımasıdır. Belli bir grubun karmařık kimliđini, o grubun 'özü' diye sunulan bir dizi basit özelliđe indirgeyerek ve bireyselliđi, farklılıđı ortadan kaldırarak grubu homojenleřtirme dölşölncesidir. Ayrıca bir dölşölnce veya sürecin basite indirgenmesidir.[4]

Antik Yunan filozofları, teolojistler, Darwin öncesi dölşölünürler tipolojik dölşölnce biçimini savunarak *tipe* odaklanmıřtırlar. [5] İlk evrim teorisyeni olarak görölen Fransız dođa bilimci Lamarck' tan (Jean-Baptiste Lamarck (1744 - 1829) "Kazanılmıř karakterlerin iletimi") önce canlıların “*ebedi deđiřmez tür*”ler olduđu dölşölünölrdü. Bu dölşölnceye göre herhangi bir canlının bir türün üyesi olması için *gerekli* ve *standart* tüm řartlara sahip olması gerekmektedir. Her řart *gereklidir* ve varlıđın türün içinde yer alabilmesi için tüm řartlara sahip olmak *standart* sayılmaktadır. Bir canlının bir türe ait olabilmesi için, tanımlayıcı karaktere gereksinimi olduđu dölşölünölür ve bu řartlar hiç deđiřmez. Eđer bir řey bir türse deđiřmez ve eđer deđiřirse bir tür olamaz. [8] Bu dölşölnce biçimine Mayr “tipolojik dölşölnce” olarak isimlendirmiřtir.

Genellikle türler tözcölük dölşölncesindeki gibi tüm kriterleri karřılayan türlere monotipik türler denmektedir. Bunlara nadiren rastlanmaktadır. Türler genellikle belirli karakteristik özelliklerinin sadece bazılarını karřılamaktadır.[8]

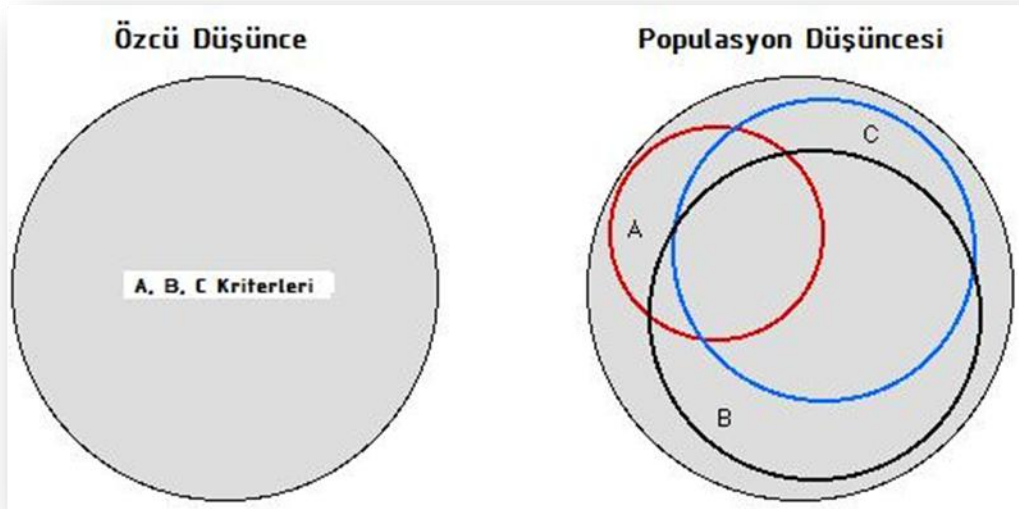
2.1.2 Popölasyon Dölşölncesi

“Popölasyon dölşölncesi” terimi de Ernst Mayr tarafından ilk olarak 1959'da kullanılmıřtır. Mayr'ın kullandıđu bu terim, Lamark, Mendel ve Darwin gibi biyoloji arařtırmacılarının teorilerinin bir devamı niteliğindedir. (DeLanda 2001) Bu terimle Mayr yeni bir řeyi

tariflemek yerine bir düşünce yolunu açmaktadır.

Popülasyon düşüncesinin varsayımları ise, tipolojistlerin tam tersidir. Popülasyoncular, organik dünyadaki her şeyin eşsizliğini vurgularlar. İnsan türü için doğru olan –ki bu iki bireyin aynı olmadığıdır – eşit şekilde tüm hayvan ve bitki türleri için de geçerlidir. Bireylerin birbirinden farkları olduğu gibi birey yaşamı boyunca kendi içinde de sürekli olarak değişmektedir. Tipolojist yaratılışçılar, bu şekilde bir değişimin ‘gerilemeyi (devolution)’ temsil ettiğini söylemektedir. Yaratılışçı ve tipçi düşünce için popülasyoncu bakış açısı, ‘saf türden’ uzaklaşan bir hareketi ima eden bir yaklaşımdır. Bir tipolojist, tür içindeki çeşitlenmeyi tür içindeki bir hata, beklenmedik sapma olarak görmektedir. Eğer dış etkilerin yoksun olduğu bir ortamda hiçbirin türün bozulmayacağını sonsuza dek aynı kalacağını düşünmektedir. Buradaki düşünce, türlerin Aristocu tözcü düşünceden gelmektedir.[8]

Popülasyon düşünürlerinin ve tipolojistlerin nihai sonuçları birbirlerinin tam tersidir. Tipolojistler için tip, gerçek ve varyasyon bir yanılısamayken popülasyoncular için (ortalama) tip, bir soyutlamadır. (Şekil 2.1) Varyasyon ise gerçektir ve üretkendir. Tipolojist düşünce fikri, Aristocu bir tündengelim yaklaşımını benimsemektedir. Benzer karakteristik özellikli varlıkların farklarını dikkate alınmaz. Bu farklar birer hata olarak görülür. Popülasyon Düşüncesi fikri ise Deleuze yaklaşımıyla, varoluşun kendisinin *farktan* gelmekte olduğunu söylemektedir.[9]



Şekil 2.1 Tip ve popülasyon düşüncesi arasındaki fark diyagramı.

Popülasyon düşüncesine göre tüm organizmalar (canlılar) ve canlı olguları eşsiz özelliklerden oluşur ve sadece istatistiksel terimlerle tanımlanabilirler. Bireyler ya da herhangi canlı varlık türü, aritmetik yollarla ve varyasyonların istatistikleriyle karar verebildiğimiz popülasyonları şekillendirir. Ortalamalar, yalnızca istatistiksel *soyutlamalardır*, ama popülasyonu oluşturan bireyler *gerçekliğe* sahiptir.(Mayr, 1976)

Popülasyon düşüncesi geniş üretken bir topluluğa ihtiyaç duymaktadır. Evrim felsefeleri Aristocu tözcülüğü yıkmaktadır. Popülasyon düşüncesi için DeLanda, zebra topluluğu üzerine bir örnek vermektedir. “Hatasız ve mükemmel bir zebranın varlığı zebra fikrinde vardır. Dünyada yaşayan zebralar ise mükemmel değildir. Popülasyon düşüncesiyle ideal zebra fikrinden *farklı* olmak önemli ve kritiktir. *İdeal olmak üretken değildir*. Çünkü homojenlik üretken değildir. Heterojen bir zebra topluluğu daha üretkendir. Doğal seleksiyon filtresiyle daha güçlü zebralar var olmaktadır.” [6] Evrim otomatik bir arama prosedürüdür. Her yerde aynı anda gerçekleşen bu aramada doğal seleksiyon bir filtre görevi görür. Bu süreç zamanla geniş üretken topluluğa yeni şeklini verdiği düşünülmektedir. (Deleuze&Guattari, 1987)

Birçok disiplin söylemlerini geliştirmek için “tipolojik özcülüğe” alternatif olacak “*Çokluk*” (multiplicity) kavramı üzerine gitmektedir. Geometriden matematiğe, biyolojiden felsefeye *çokluk* konsepti aynı gerçeklere bakmak için farklı bir yoldur. Deleuze; “*Çokluklar (Multiplicities)* kitlelerin bir kombinasyonu veya bir “tek”i işaret etmek zorunda değildir daha çok bir sistem olmak için herhangi bir bütünlüğüne ihtiyaç duymayan kitlelerin organizasyonudur” demektedir. (Deleuze, 1994)

“Çokluk”lar, sadece kategorizasyon ve tipleştirmenin karşısı değildir. Özünde bir varlık felsefesidir.[9]

Popülasyon düşüncesi günümüzde işleyen tipolojik düşünceye alternatif olabilecek bir bakışı araştırma girişiminde bulunmaktadır. “Popülasyon düşüncesi” tabanlı kavramlar olarak ise *kolektif zeka, kendini organize eden sistemler, evrimsel sistemler, karmaşık sistemlerden* söz edilebilir. Ayrıca insan zihni ve tasarım süreci kompleks fenomeni içinde değerlendirmektedir.

Popülasyon düşüncesi konsept bir kod ile özde ve fonksiyonda aynı olan fakat kişiye özel şekillendirilebilen farklı ürünler üretmesine yol açmaktadır. Dijital tasarım kavramı olan, kitlesel bireyselleştirme (Mass Customization) popülasyon düşüncesiyle filizlenmektedir.

2.2 Özgöl (İçlem) Düşünce

Popölasyon düşünçesine Deleuze'un eklediğı, bugünkü şeklini termodinamikten alan başka bir algılama biçimi vardır ancak bunun kökenleri Leibniz felsefesine kadar dayanır ki bu da "Özgöl Düşünce"dir (Intensive Thinking). Türk Dil Kurumu sözlüğünde bu kavrama karşılık gelebilecek olan kavram "içlem"dir. İçlem, "bir kavramın çağrıştırdığı kapsama giren niteliklerin veya taşıdığı özelliklerin bütünü, tazammun: "Kuş sözcüğü bize canlı, havada uçan ve öten bir varlık anlatır; şu halde canlılık, uçuculuk ve ötüclük kuş kavramının içlemine giren niteliklerdir. Kartal, kırlangıç, tavuk ve öbür kuşlar ise birer kuş cinsi olduklarından bunun içlemini oluştururlar." veya bir nesnenin içeriğini oluşturan şey", olarak geçmektedir.[1]

Özgöl olmayan nicelikler, mimarların çok aşına olduğu uzunluk, alanlar, hacimler gibi büyüklükleri ifade eder. Bunlar, mekansal olarak tekrar bölünebilen büyüklükler olarak tanımlanır örneğin eğer biri bir bardak su alır ve ikiye bölerse sonuçta iki farklı hacimde su elde edilir. Diğer yandan "özgül" terimi, ısı, basınç, hız gibi tekrar bölünemeyen çokluklar olarak tanımlar. 90 derecedeki bir hacim suyu ikiye bölündüğünde, 45 derecede iki hacim su elde edilmez, 90 derece sıcaklıkta iki hacim su elde eder. Deleuze için bölünebilirliğin olmaması önemli olsa da *özgül* niceliklerin başka özelliklerini vurgular: "özgül"lük bir farklılığı eş zamanlı olarak süreçte kendini dengeleme eğilimindedir; Bu maddenin ve enerjinin akışını sağlar. Başka bir deyişle, "özgül"lük "*fark*"lılıkları üretken farklılıklardır. Örneğin, döllenmiş yumurtadan insan vücudu üreten embriyo süreci, "özgül"lük farklılıkları tarafından yürütölen bir işlemdir (kimyasal konsantrasyon, yoğunluk ve yüzey basınçlarının farklılığı gibi).(DeLanda 2001)

Deleuze'un "*Özgöl düşünce*"si Leibniz'in "*Monadoloji*" çalışmalarından etkilenmiştir. Monadoloji terimi Gottfried W. Leibniz'in "Monadology" isimli eserinde açıklanmaktadır. Monadoloji terimi Latince "monad" teriminden gelmektedir. Eski yunanca "monos" ya da "mono" yani "tek" anlamından türetilmiştir. Tek, Leibniz'in çalışmalarında "bölünemez" ya da "öz" anlamında kullanılmaktadır. Leibniz her şeyin bölünemez "*monad*"lardan oluştuğunu savunur.[7]

DeLanda, mimari açıdan düşünüldüğünde, tanımlanan bir mimari eleman, binanın geri kalanı ile felsefi ve matematiksel olarak birbirine bağlanmazsa bina gerekli işlevini yerine getiremeyebilir, demektedir. (DeLanda 2001)

Konsept terimi ise fikrin soyutlanması ya da tipik olarak bir dil veya sembololojiyi temsil eden zihinde uyanan semboldür. [2] Özgül terimi, mimarların herhangi bir tasarım ürününü ortaya koyarken dillendirdiği “*konsept*” terimi ile örtüşmektedir. Biyolojide “*gen*” terimini de karşılamaktadır. Bilgisayar bilimlerinde ise “*algoritma*” terimi özgül terimiyle üst üste gelmektedir.

Günümüzde bu konu yeni bir yaklaşımla bilgisayar algoritmalarının matematiksel bir konsept veya matematiksel bir genetik oluşturulması olarak algılanabilir. Bilgisayar kodları, parametreleri doğru verildiğinde mimara geniş üç boyutlu uzayda yeni tasarım alanları açmaktadır.

2.3 Topolojik Düşünce

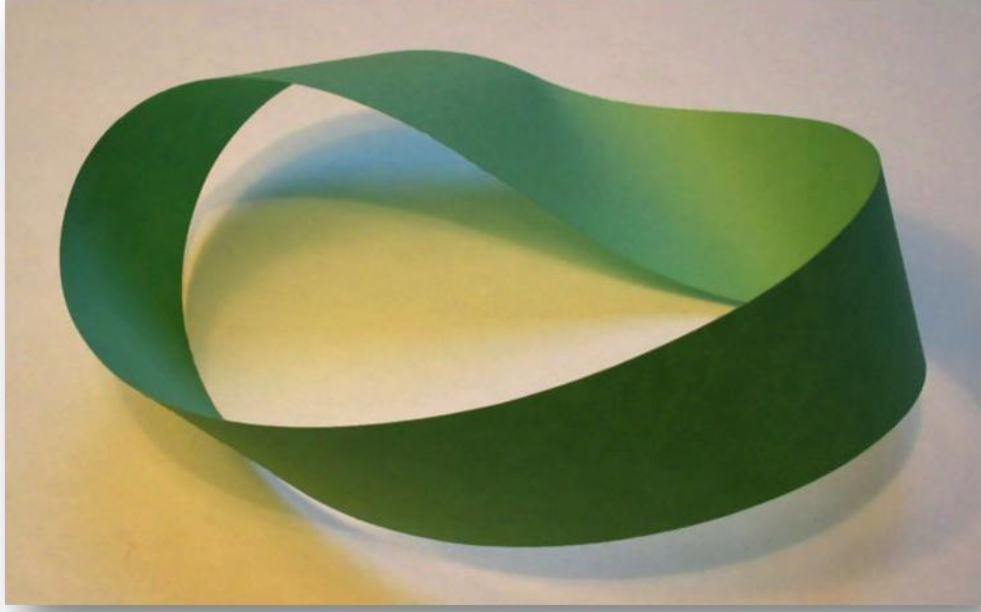
Topoloji, Eski Yunanca 'da yüzey veya uzay anlamına gelen *topos* ve *bilim* anlamına gelen *logos* kelimelerinden türetilmiştir. Dolayısıyla, topoloji, uzaylar veya yüzeyler bilimidir. [2]

Topoloji geometrisinde amaç, nesneleri yırtmadan ve koparmadan, eğip bükerek bir başka nesneye dönüştürebilmektir. Topoloji, matematiğin bir dalı olarak 19. yüzyılın sonlarında ünlü Fransız matematikçi Henri Poincaré'nin (1854–1912) çalışmaları ile sistematik oluşumuna başlamıştır. Aslında, topoloji alanındaki araştırmaların başlangıcı olarak, G. Riemann'ın (1826 - 1866) 19. yy. ortalarına rastlayan fonksiyonlar teorisi ile ilgili çalışmaları alınabilir. Buna rağmen, ilk topolojik kavramları ortaya atıp üzerine derin bir teoriyi kuran Poincaré'dir. Poincaré : “Topoloji, geometrik şekillerin, sadece alışılmış uzayda değil, üçten fazla boyutlu uzaylarda da niteliklerini öğrenmemizi sağlayan bir bilimdir,” demiştir.[11]

Bir geometrik şeklin niteliksel özelliklerini daha iyi anlamak için bir küreyi, elastik bir balon olarak düşünelim. Bu küreyi, parçalamadan veya farklı iki noktasını yapıştırmadan şeklinde meydana getirilebilen tüm değişimlere *homeomorfizma* denir. [11] Bunun için de "homeomorfizma" adındaki denklik bağıntısı tanımlanmıştır. Latince *homois* = benzer ve *morphe* = form kelimelerinin birleşiminden oluşmaktadır. [2] Homeomorfizma, topolojik denkliktir ve uzayları denklik sınıflarına ayırır. Başka bir deyişle X ve Y uzayları homeomorf ise X ile Y aynı topolojik özelliklere sahiptir, denebilir.[10] O halde, kürenin niteliksel özellikleri, tüm homeomorfik taklitleri ile paylaştığı özellikleridir. Bu özelliklere de *topolojik özellikler* denir.

Homeomorfizmaya örnek olarak, bir üçgenin bir çembere ya da bir çay bardağının, çay

tabağına dönüşümünü alabiliriz. Bunu geometrik olarak görmek çok kolaydır. Gerçekten çay bardağı ya da tabağından birinin kauçuktan yapıldığını düşünürsek, o cismi yırtmadan, kesip koparmadan sadece çekip uzatarak ve eğip bükerek diğer cisme dönüştürebileceğimizi görürüz. Benzer şekilde kulplu bardak ve simidin birbirlerine aynı yöntemle dönüştürülebileceğini de görebiliriz. [10] Topoloji geometrilerinin en popüler olanı Alman matematikçi A. Ferdinand Möbius 1858 yılında ortaya koyduğu Möbius bantıdır (Şekil 2.2)



Şekil 2.2 Möbius Bantı

Poincaré'in dediği gibi topoloji N boyutlu uzay bilimidir. N boyutlu uzaylarla uğraşır. N boyutta uzayın her elemanın komşulukları ve bu komşuluklarla noktanın cebirsel özelliklerini irdeler. George Friedrich Bernhard Riemann (17 Eylül 1826 - 20 Temmuz 1866) ve Johann Carl Friedrich Gauss ya da Gauß (30 Nisan 1777 – 23 Şubat 1855) analiz ve diferansiyel geometri dalında çok önemli katkıları olan Alman matematikçilerdir. [2] Söz konusu katkılar daha sonra Einstein'ın ünlü izafiyet teorisinin geliştirilmesinde önemli rol oynamıştır. 19. yüzyılda Gauss ve Riemann tarafından, matematikte Kartezyen uzayına alternatif olarak diferansiyel denklemleri ve diferansiyel geometri fikrini üretilmiştir. Bu geometri ve fonksiyonlar topolojik uzayda kullanılmaktadır. [6]

Diferansiyel denklemler, matematikte, fonksiyon veya fonksiyonların, bir veya birden çok

değişkene göre *türevlerini* ilişkilendiren denklemlerdir. Fizik, kimya, mühendislik, biyoloji ve ekonomi alanlarında matematiksel modeller genellikle diferansiyel denklemler kullanılarak ifade edilirler. [2] Fonksiyonların türevlerinin ilişkilendirilmesi fonksiyonların değişim oranlarına odaklanılmasını sağlamaktadır. [6] Örneğin, bir kütlenin hızlanması sürecini, zaman-hız fonksiyonuyla değil, zaman-ivme fonksiyonu ile değerlendirilmesi gibidir. İvme anlık hız ile ilgilenmek yerine anlık hız *farkıyla* ilgilenmektedir. Anlık *oluş* değeri değil *oluş* değeri *farkları* üzerine çalışılır. İki farklı zamanda iki değer arasındaki ilişkinin fonksiyonlaştırılmasıdır. Bu yolla “*oluş farkının hızı*” değerlendirilmektedir. Ortaya “*oluş farkı uzayı*” çıkmaktadır. Kütle Kartezyen uzayda hareket halindeyken eğer kütlenin hızında bir değişim yoksa oluş farkı uzayında sabit kalmaktadır.

Diferansiyel denklemler, günümüz dünyasının dinamiklerini modellemeye yaramaktadır. Deleuze varlık felsefesi farklılıklar ve farklılıkların ortaya koyduğu üretkenlikten bahsetmektedir. Diferansiyel geometri farklar arasındaki *ilişkiyi* matematiksel olarak ortaya koyulmasına yardım etmektedir. Gauss ve Riemann çalışmalarıyla Kartezyen uzayında çözülemeyen *ilişkileri* n boyutlu uzayda diferansiyel geometri yoluyla çözmüştür. [6]

Sonuçta topolojik düşünce, “oluşun, anlık farklarının değerlendirilmesi” fikrine işaret etmektedir. Günümüzde mimari tasarımlarda sıklıkla kullanılmaya başlayan topolojik geometriler aslında Poincaré, Gauss ve Riemann gibi matematikçilerin ortaya koyduğu bir sistemin alt ürünüdür. Dinamik bir oluşun, anlık farklarının matematiğe dökülmesidir. Deleuze, geometriden aldığı terimi soyutlayarak, topolojik düşünceyi kurduğu sisteme eklemiştir. Özgül ve popülasyon düşüncesiyle ortaya koyduğu fikirlerin birbirleriyle ilişkilerinin topolojik olması gerekliliğini vurgulamaktadır. Varlığı fark üzerinden değerlendiren, varlıkların bu farklarla özgün olacağını savunan Deleuze farklılıkların birbirleriyle ilişkilerini de topolojik olarak değerlendirmektedir.

Sonuç

Tüm bu kavramlar bir araya getirildiğinde, dijital tasarım için bir bakış açısı gelişmektedir. *Topolojik düşünce*, tasarım ürünün özündeki düşüncenin (kodun, kavramın, konseptin) esnek, eğrilebilir, dönüştürülebilir ve sürdürülebilirliğine de vurgu yapmaktadır. *Topolojik düşünce*, tasarım ürünün özünün, *popülasyon düşüncesindeki* gibi farklılıkları vurgularken esnekliğini; *özgül düşüncesindeki* gibi bireyselliğini korurken, kodun tasarımın geri kalanı ile felsefi ve

matematiksel olarak birbirine bağlanmasını sağlamaktadır. Topolojik olarak üretkenliği sağlamak için ise “özgül fark”ın vazgeçilmez olduğu fikri ortadır.

Sayısal mimarlık, fikir altyapısını özellikle Deleuze üzerinden kurmaktadır. Deleuze’ün varlık felsefesi kavramları dijital tasarım sürecine yol göstermektedir. Özellikle **popülasyon düşüncesi, özgül düşünce, topolojik düşünce** bilgisayarlı tasarım sürecinin yönünü göstermektedir. Bu üç kavram sırasıyla **fark (different), uygun (fit) ve ilişki (relationship)** kavramları üzerine gitmektedir.

Bu üç ana fikrin yanı sıra, Deleuze’ün bu fikirlerinin getirdiği diğer alt kavramlarda bütünü tamamlamaktadır. Bu kavramların yanı sıra ek alt kavramlar **Doğrusal olmayan** (Non-Linear), **bağlar arası** (inter-connectivity), **süreklilik** (continuity), **iletişim ağı** (networks), **dinamizm** (dynamism), **şema** (diagram), **tasarım makineleri** (design machines), **hiper-bağlar** (hyper-connectivity), **Hiyerarşik olmayan strüktürler** (non-hierarchical structures) Deleuze fikir ve teorilerinin içinde var olan ve bu üç düşünce biçimini algılamamızı sağlayan kavramlardır. Dijital tasarım sürecinin **doğrusal** ve **hiyerarşik olmayan, dinamik** bir **sürekliliğe** sahip olduğu dile getirilmektedir. Hiper bağlar, bağlar arası ve iletişim ağı kavramları ise özellikle **ilişkileri** kurgulamaktadır. Bu kavramlar, tasarımcı, tasarım ve bunların ilişkilerini, hiyerarşik olmayan bir biçimde ortaya koymaktadır. Bu ilişki sisteminin oluşması için kullanılabilecek araç, özellikle internet olmaktadır.

İnternetin anarşik ağ sistemi ve bunun yarattığı sosyal ve kültürel değişim mimarların fikir yapılarını da etkilemektedir. Deleuze’ün kullandığı kavramlar içinde var olan kavramlar tasarımcıya mikro ve makro ölçeklerde yeni bakış açıları sunmaktadır. Özellikle wiki kavramıyla ortaya konan siteler, kolektif zekaya sahip kendini organize eden bir sistem olarak adım adım değişerek gelişmektedir.

Daha önce de belirtildiği gibi; bu kavramlar dijital tasarım fikrinin zeminini örmektedir. Bu zemin üzerine, yine bilimsel ve mimari kavramlar, birbiriyle **hiyerarşik ve doğrusal olmayan, dinamik, açık uçlu** bir örüntü oluşturmaktadır.

Zamanımızda genellikle kullanılan “disiplinler arası iletişim” kavramı popülasyon düşüncesiyle irdelendiğinde, “disiplinler arası” kavramı burada yetersiz kalmaktadır. Popülasyon düşüncesinin altında mimarlar için “disiplin ötelemesi” yatmaktadır. [9] Mimarlar, disiplinler arası kombinasyonu aramaktan çok diğer disiplinlerin konseptlerini,

kelime dağarcıklarını, tekniklerini iyice inceleyerek, bunlar ile mimarlık kavramlarının sınırlarını genişletmek ve mimarlık disiplini diğer disiplinlere doğru ötelemek için çalışmalıdır.

Mimarlık disiplini çeşitli disiplinler ile iletişindedir. Bu sayede tasarımlarını uygulanabilir hale gelmektedir. Sayısal mimarlık ise, sürecin sayısallaştırılması çabasına destek olacak disiplin olarak bilgisayar bilimcilerini görmektedir. Bilgisayar bilimcilerinin kavramlarını algılamak ve gerektiği kullanabilmek için mimarlık disiplini, alanını öteleyerek genişletmektedir. Mimarlar bahsi geçen öteleme ile dijital tasarım sürecinde, tasarımcı makineye ne yapması gerektiğini söyleyen olmaktadır. Bilgisayar desteği mimarların tasarım ve inşa sürecindeki kontrollerini güçlendirecektir.

Sayısal tasarım fikir altyapısında özellikle Deleuze'ün pozitif bilimlerin kullandığı terimler yardımıyla ortaya koyduğu varlık felsefesinin derin izleri görülmektedir. Bununla birlikte, pozitif bilimler de, sayısal tasarım fikrini doğrudan etkilemektedir. Özellikle matematik, bilgisayar ve biyoloji bilimlerinin üzerinde durduğu parametre, algoritma, algoritmik düşünce, kompleks sistemler kendini organize eden sistemler, evrim ve bunların matematiksel açılımları dijital tasarımın ilgilendiği alanlardır. Bu bölümde fikir altyapısı üzerinde durulan sayısal tasarım, üçüncü bölümde özellikle pozitif bilimler ve mimarlık disiplini bağlamında incelenecektir.

3. MİMARİ TASARIM VE ALGORİTMİK DÜŞÜNCE

Günümüzde paket programlara, tasarımcının ihtiyacı doğrultusunda eklemeler yapmak (script yazmak) sıradan bir iş haline gelmektedir. Mimarlık disiplininde bu değişimi yakından son yıllarda dikkatle izlemekte ve deneysel ürünler vermektedir. Konu sadece paket programların isteğe göre değiştirilmesiyle sınırlı değildir. Konu, altyapısında bir düşünce biçimini de beraberinde getirmektedir.

Tasarım kavramı kompleks kavramıyla örtüştürülmektedir. Tasarım sürecinin de, diğer kompleks sistemlerde olduğu gibi, çözümlenemeyecek kadar çok girdisi olduğu açıktır. Günümüzde kompleks sistemler üzerine yapılan çözümleme çalışmaları algoritmik düşünce baz alınarak yürütülmüştür. Bu sebeple bu bölümde *tasarım* kavramı ve *kompleks sistemlerin* çözümünü öngören *algoritmik düşünce* sisteminin üzerinde durulacaktır.

Yazılım üretmek veya bir yazılımı manipüle edebilmek için, sadece yazılım dilini bilmek yeterli değildir. Bilgisayarın çalışma mantığını bilmek gerekmektedir. Bilgisayarın donanımının çalışma mantığından çok, bilgisayar yazılımlarının getirdiği, fikir alt yapısını algılamak önemlidir. Bu fikir algoritmik düşüncedir. Algoritmayı anlamak, sayısal tasarım için kritik önem taşımaktadır.

3.1 Algoritma ve Algoritmik Düşünce nedir?

Arap matematikçi Muhammed ibni Musa El-Harezmi (Al-Khwarizmi) (780?-850?), isminden batıya daha sonra Fransızcadan Türkçemize gelen “algoritma” Türk Dil Kurumu sözlüğünde (www.tdk.org.tr) “belirli herhangi bir kurala bağlı bulunan her türlü hesap işlemi, Harezmi yolu” olarak bulunmaktadır. The Oxford English Dictionary içinde algoritma, “bir işlem veya kurallar dizisi, bir matematiksel ifadeleme dizisi, şimdi genellikle bilgisayar ve makine dillerinde kullanılmaktadır,” olarak geçmektedir. Türk Dil Kurumu sözlüğünde, “parametre” ise “Cebirde bir denklemin kat sayılarına giren değişken nicelik.” olarak belirtilmiştir.[1]

Algoritma matematiksel (veya cebir diyebiliriz: cebir Arapça kökenlidir) denklemler dizini ise her algoritma bir veya daha fazla parametre içerebilmektedir. Algoritmadaki her kuralın, içerdiği parametrelere bağlı bir sonucu vardır.

Algoritma belirli bir problemi çözmek için bir formül ya da adımlanmış kurallar setidir. Bir algoritmanın kesin kurallar seti ve açık bir bitme noktasının olması gereklidir. Algoritmalar her tür semantikte ifade edilebilir. Bu semantikler İngilizce, Fransızca, Türkçe gibi doğal

dillerden BASIC, C gibi programlama dilleri olabileceği gibi mimarlık teknik dili olabilmektedir. Algoritmalar gün içinde her an kullandığımız bir araçtır. Bir kek tarifi bile bir algoritmadır. (Harel, D. 1992)

Algoritmik düşünce; bütün ve karışık bir problemin, küçük birimlere bölünerek ve bu birimlerin yaptığı basit işlemlerin ortaya konarak, bütün içindeki yerlerinin belirlenmesi için kullanılmaktadır. Bütün bir sistemi formüle etmek yerine birimlerin yaptığı daha basit işlemlerin çözümlenmesine gidilmektedir. Birim otomatların basit davranışları algoritmik düşünce için önemlidir. Algoritmik düşüncenin ana fikri, fonksiyonel ayrıştırma, kendini tekrar etme, basit bilgi organizasyonu, parametrikleştirme, kavramlarını içerir. Her algoritmik düşünce tipinde ileri matematik kullanılmasına gerek yoktur. (Harel, D. 1992)

Gerçek hayatımızda farkında olmadan uyguladığımız bu düşünce biçiminde, en önemli nokta kuralların açık olmasıdır. Örneğin bir kek tarifinde; hamurun “kulak memesi kıvamına gelmesi” sübjektiftir. Bunun yerine hamurun içeriğinin hangi malzemelerden oluştuğu, bu malzemelerin kaç gram veya mililitre konulması, gerektiği kaç dakika, kaç Newton kuvvet uygulanarak yoğrulması gerektiği ve hatta ortam sıcaklığı verilmelidir. Çünkü bir algoritmanın işlemlenmesinde sağduyu faktörü yoktur. Kendini tekrar etme ve parametrikleştirme bu açıdan önemli olmaktadır.



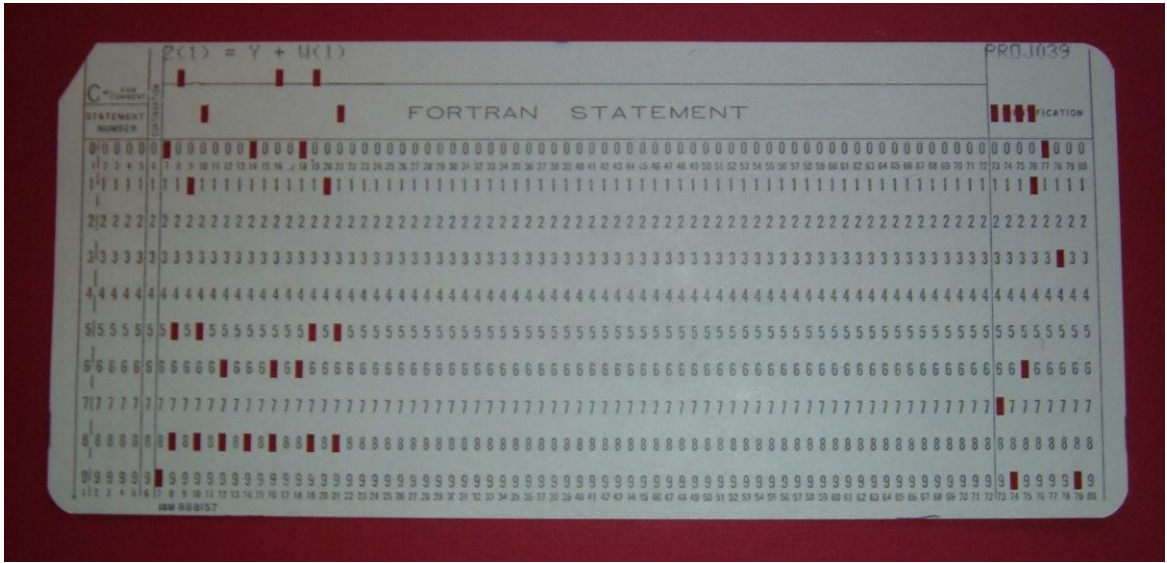
Şekil 3.1 Abaküs



Şekil 3.2 Jacquard Tezgahı

Algoritmik düşünce fikri yeni bir buluş değildir. Kökleri El-Harezmi'nin matematik teorilerine kadar uzanmaktadır. Abaküs (Şekil 3.1), Leibniz Hesap makinesi ve Jakar Tezgahı (Şekil 3.2) (Joseph Marie Jacquard (1752 – 1834)) bu soyut fikirle üretilmiştir. Özellikle Jakar'ın dokuma tezgahında kullandığı fikir, algoritmik düşüncenin bir yansımasıdır. Jacquard, nümerik olarak kontrol edilebilen ilk makineyi, delikli kartlarla dokunacak desenin kontrol edilebildiği mekanik dokuma tezgahını, icat eden Fransız mucittir.

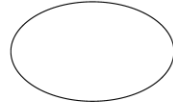
1801 yılında Jacquard tarafından üretilen bu mekanik algoritma fikri, yakın tarihe kadar bilgisayar ile iletişim kurmak için kullanılan, delikli kart (Şekil 3.3) mantığında kullanılmıştır. Bu kartlar algoritmik düşüncenin bir ürünüdür. Bir fikir veya Jacquard tezgahında üretilecek bir patern “evet, hayır”, “açık, kapalı”, “1, 0”, “delik, delik değil” gibi basit yönergelerle adımlanarak kurgulanmaktadır. Basit yönergelerin şematize edilerek ortaya konması için ise akış diyagramları kullanılmaktadır.



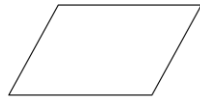
Şekil 3.3 Delikli kart

Akış Diyagramı

Akış diyagramı algoritmik düşüncenin sekanslar halinde algılanmasını sağlayan grafik ifadeleme (Şekil 3.4) biçimidir. Semboller seti ve kısa metinlerden oluşmaktadır. Binlerce bileşeni olan bir algoritma bu belirli sayıdaki sembol ve yazılarını dizimiyle anlatılabilmektedir.



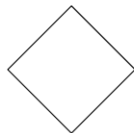
Oval - Başlama ve bitirme kutuları



Paralelkenar - Bir girdi veya çıktı



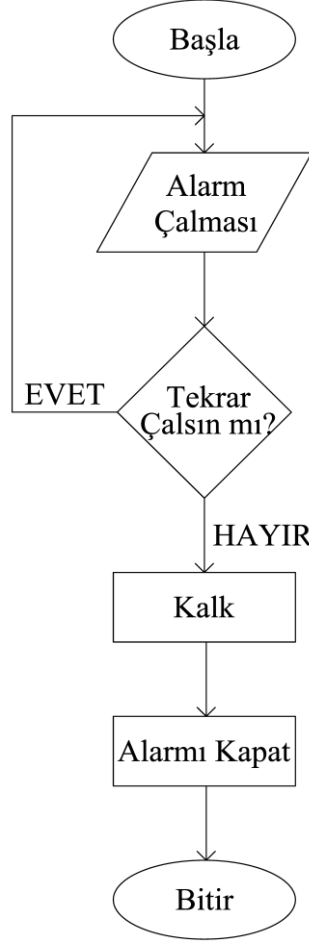
Dikdörtgen - Eylem belirten kutular



Karo - Karar ya da soru kutuları

Şekil 3.4 Akış Diyagramı hazırlarken kullanılabilecek şekiller

Bir butona basarak alarmin ertelendiği sabahları kalkma işlemini algoritmik bir akış diyagramına (Şekil 3.5) dökersek şu şekilde olur:



Şekil 3.5 Örnek Akış Diyagramı

Bazı problemler algoritmaya indirilemeyecek kadar karmaşık sistemlere sahiptir. Global ekonomi, iklimler, sinir sistemleri, telekomünikasyon ağları, hücreler, insan zekası ve yaşayan organizmalar gibi *karmaşık sistemler* soyutlanamaz veya adımlanamaz karmaşıklığa sahiptir. Karmaşık sistemlerin algoritmaya dökülememesinin sebebi algoritmanın tanımında verilen “kesin ve açık kurallar” günümüz teknolojisinde ortaya konulamamasıdır. İnsan zekası ve tasarım süreci karmaşıklık kavramıyla birlikte anılmaktadır. *Karmaşık sistemler* henüz algoritmaya dökülmese de bir *kolektif zeka* ile işlemekte oldukları görülmektedir. Kolektif zeka ve kompleks sistemler, ikinci bölümde *monad* ve *özgül düşünce* fikirlerinde bahsedilen, en alt birimlerin kolektif davranışlarıyla ortaya konan sistemlerdir. Adı geçen sistemler, çoğunlukla birimlerin kolektif olarak “iş”lemesiyle çalışmaktadır. Birimlerin *çokluğu* ve her

birimin problem içindeki her birimin sistem içindeki parametrelerinin *farkı* sebebiyle, karmaşık sistemler henüz algoritmaya dökülememişlerdir. Bu durum bu sistemlerin tamamının veya bir kısmının gelecekte algoritmik bir yapıyla çözümüne engel olmamaktadır. Kompleks sistemlerin çözümlenmesi amacı güden araştırmalar çözümlemenin algoritmik bir biçimde olacağını öngörmektedir.

3.1.1 Kompleks Sistemler, Kendini Organize Eden Sistemler, Kolektif Zeka

Matematikçi Stephen Wolfram “Tüm süreçler ister insan çabasıyla ister doğada kendiliğinden gelişsin hesaplamaya yönelik işleme dökülebilir” demektedir. (Chu, 2006) Kompleks sistemler hassas dengeler üzerinde duran en küçük değişimlere beklenmedik sonuçlar verebilen sistemlerdir. Kompleks sistemler genellikle *kendini organize eden* ve *olagelen* kavramlarıyla özdeşleştirilmektedir. Bir kompleks sistem genellikle *kolektif zeka* örneği olarak göstermektedir. İnsan zekası ve tasarım süreçleri kompleks kavramıyla değerlendirilir.

Kompleks Sistemler

Kompleks sistemler, doğrusal olmayan ve başlangıçtaki koşullara karşı duyarlı olan ve bu koşulların küçük değişiminde bile küresel ölçekte büyük çalkantılar oluşturabilen sistemlerdir. Şehirler kompleks sistemlerdir. İçerisindeki insan ve araç akışıyla şehirler bir “olagelen” davranış sergilemektedir. Çünkü içindeki her birim için çok sayıda kararın aynı anda birbiri ile iletişim halinde alınmasını gerektirmektedir. Bu tarz kompleks sistemleri çözümlemenin iki tür yolu vardır. Basit ve hızlı yolu, var olan bir sistemin davranışlarını inceleyerek matematiksel tanımlara soyutlamaktır. Ardından bu soyutlamayı gözlemlenebilir paternlere dağıtarak parametrik bir modelini oluşturmaktır. Şehirlerde trafik akışını akışkanlar dinamiğinin yardımıyla soyutlayarak parametrik bir modelini oluşturmak gibidir. Hız ve yoğunluk parametreleri ile modellenen trafik ağı daha sonra oluşabilecek sorunlara karşı davranışları model üzerinde görülebilir. (Stathopoulos, Weinstock, 2006)

Kendini Organize Eden Sistemler

“Kendini organize etme” içsel bir işlemdir. Sistemin kendini, çevre etmenlerine karşı dışarıdan yönlendirme ve yönetme olmadan özel bir fonksiyon için düzenleme işlemdir. (Hensel, 2006a)

Termit, sayıları milyonları bulan koloniler halinde yaşayan eklembacaklı canlılardır. Ağızlarındaki salgıyı toprakla karıştırarak sert ve dayanıklı bir malzeme elde etmektedirler.

Yuvalarını 6 metre yüksekliğinde, 12 metre genişliğinde yapabilmektedirler (Şekil 3.6). Yuvalarının içine hava çevrimini sağlamak için özel koridorlar yapıp oluşturdukları gözeneklerle havayı filtre edebilmektedirler. Her termit türü bulunduğu ortamın koşullarına uygun yuvalar yapmaktadırlar. Yağmurlu bölgelerde yaşayan bazı termitler, özel çıkıntılı çatısı olan mantar benzeri tepeciklerden oluşan yuvalar inşa ederler. Kimi türleri yuvalarını dünyanın manyetik alanına göre şekillendirirler. Büyük sahra'da yaşayan türleri ise zeminin 40 m. kadar aşağısına kadar inerek, yukarıdaki yuvaya suyun buharlaşarak ulaşmasını sağlarlar. Yuvaların kalın duvarları ise içerideki nemin korunmasını sağlamaktadır. Bu canlılar bu kompleks sistemi basit adımlarla kendilerini tekrar ederek ve doğal parametreleri (nem, ısı, manyetizma gibi) değerlendirerek hareket ederler. Bu sosyal sistemi birçok açıdan değerlendirebiliriz. Sistemin her bir biriminin algoritmik olarak basit kurgularla hareket etmektedirler. Sistem incelendiğinde, birim otomatların (bir termitin) basit yönergelerle, var olan sistemin devamını sağladığı görülmektedir. *Açık uçlu, üretken ve “olagelen”* bir sistem gözlenmektedir.



Şekil 3.6 Termit kuleleri

Sonuç olarak kompleks sistemler, kendini organize eden sistemler ve kolektif zeka kavramları birbirlerinin içine girmiş kavramlardır. Bu kavramların üzerinden geçerken asıl amaçlanan bu sistemlerin altında yatan ana fikirlerin “açık uçlu, kendini tekrarlayan, parametrik, üretken ve olagelen” olduğunu söyleyebiliriz.

Bahsedilen sistemlerin üzerine yapılan çalışmalarda, bu sistemlerin çözümleri için algoritmalar kullanılmaktadırlar. Algoritmalar termit canlısında olduğu gibi basit yönergeler kullanarak kompleks sistemler üretebilmektedir. Kompleks bir sistem üretebilen algoritmalar, var olan kompleks sistemlerin çözümü içinde kullanılabilir. Deleuze’un de ortaya koyduğu soyut makine fikri ve bilgisayar bilimlerinde kullanılan evrensel makineler kavramı, algoritmik düşünce ile üst üste gelmektedir.

3.1.2 Evrensel, Soyut makineler ve İşleme (Computation)

İnsan günümüzdeki birçok teknolojiyi doğayı taklit edip, soyutlayarak üretmiştir. Bahsi geçen kavramların soyutlaması özellikle yakın geçmişte bilgisayar ve matematik bilimlerinde kullanılmıştır. Bu soyutlama ile “evrensel makineler” üretilmiştir. Algoritmik düşüncenin gelişiminde yardımcı olan “evrensel makineler” ya da Deleuze’un deyimiyle “soyut makineler” algoritmik düşüncenin altyapısını oluşturmaktadır. Deleuze “Soyut bir makine fiziki ve maddi değildir, bunun ötesinde simgesel ve diyagramatiktir. Madde ile değil durumla, form ile değil fonksiyon ile işler. Diyagramatik ya da “soyut makine” reel olanın temsilini buna nazaran oluşmakta olan yeniyi, yeni bir gerçekliği inşa eder” (Deleuze, 1994) demektedir. Bu tanıma uyan soyut makineler fikri, bilgisayar bilimlerinde “evrensel makineler” olarak adlandırılırlar. Bu makinelerde doğal dillerin üzerinde daha soyut bir dil kullanılmaktadır. Madde ile değil o anki durum ve *oluşuyla* ilgilenilmektedir. Form değil fonksiyonlar ön plana alınmıştır. İkinci Dünya Savaşı ve sonrasında, günümüzde bilgisayar ve matematik bilimlerinin mihenk taşı kabul edilen bazı evrensel makineler üretilmiştir. Üretilen bu soyut makineler genellikle üreticisinin adıyla birlikte anılmaktadır. Bilgisayar bilimlerinin babası sayılan Alan Turing tarafından üretilen, Turing Makinesi (Turing machine), Neumann’ın Evrensel Kopyalayıcı ve Üreticisi (Universal copier and constructor), Biyolog Aristid Lindenmayer’in “L-sistemi”, matematikçi John Conway’ın “Yaşam Oyunu” (The Game of Life) evrensel makineleri özellikle bilgisayar biyoloji ve matematik bilimlerinde hala tartışılmaktadır. Bu soyut makineler kompleks bir sistemi çözümlemeye yönelik çalışmalar üzerine ortaya çıkmıştır. Soyut makineler sınırlı sayıda iş ya da eylem ile basit çekirdek birimlerin aynı anda

işlemlenmesiyle ortaya kompleks birer sistem çıkmaktadır. Kompleks sistemler üreten basit otomatlar, diğer kompleks sistemlerin de çözüm önerisi olarak görülmektedir.

Turing'in makinesi (1936)

Alan Mathison Turing (1912 – 1954), İngiliz matematikçi ve bilgisayar bilimcisidir. Turing makinesi, son derece basit, soyut, sembol işleme aracıdır. Aracın basitliğine rağmen yapılagelmiş her türlü bilgisayar mantığının kurgulanmasında bu soyut makineden yararlanılmıştır. Sadece sınırlı sayıda “iş” ile (oku, yaz, sağa git, sola git) bu evrensel makine kendini kopyalayarak organize olmaktadır. Turing bu makinesiyle pratikte fiziksel bir bilgisayar teknolojisi üretmemiştir. Bir soyut algı biçimi organize etmiştir. Günümüzde hala Turing'in yapmış olduğu bu “*evrensel makine*”, kompleks sistemler ve bilgisayar mantığı dallarının ana fikrini oluşturmuştur. Turing bu makineyle mekanik bir prosedür ya da algoritma fikri üretmiştir. (Rocker, 2006)

Neumann'ın “Evrensel kopyalayıcı ve üreticisi” (1940)

John von Neumann (1903 – 1957) Amerikalı matematikçi ve bilgisayar bilimcisidir. John von Neumann'ın “Evrensel kopyalayıcı ve üreticisi” (Universal Copier and Constructor) kendini kopyalayan bir “hücreli otomat” makinesidir. Bilgisayar kullanılmadan 1940'lı yıllarda tasarlanmıştır. 1940'da henüz DNA strüktürünün her hangi bir modeli keşfedilmemiştir. Bu açıdan bu sistemin ortaya koyduğu açık uçlu sistematik evrim mutasyon ve üreme günümüzde birçok bilgisayar ve matematik bilimi konusunun temellerini oluşturmaktadır. 29 “iş” üzerinde çalışan bu sistem, teorik biyoloji ve yapay zeka dallarında açık uçlu, üretken evrimsel sistemlerin bir soyutlaması olarak görülmektedir. [2]

“L-sistem” (1968)

Macar biyolog Aristid Lindenmayer (1925 - 1989) basit çok hücreli organizmaları incelemiştir. Lindenmayer basit çok hücreli organizmaların gelişimini bilgisayar bilimcilerinin yardımıyla soyutlayarak modellemiştir. L-Sistem olarak anılan bu model iki prensip üzerinden çalışmaktadır. Birinci prensip analitik diğeri ise üretken prensiptir. Analitik prensip sistemin üyelerini belirlerken üretken prensip ise bir algoritma yardımıyla sisteme yeni üyeler üretmekte ve sistem prensiplerini yeni oluşan sisteme göre yeniden yazmaktadır. Bu bir bitki büyümesinin soyut modellemesidir. Kendini organize eden ve kendini tekrar ederek gelişen bir sistemdir. Günümüzde de L sistem ana fikri mimarlık disiplinde yarı

otonom sistemlerin algoritmalar yardımıyla ortamın ekolojisine uygun tasarımlar üretilmesi açısından incelenmektedir. (Hensel, 2006a)

Conway'in “Yaşam Oyunu” (1970)

John Horton Conway (1937 -), İngiliz matematikçi ve bilgisayar bilimcisidir. Yaşam Oyunu (The Game of Life) Conway tarafından tasarlanan bir hücresel otomat makinesidir. Bu soyut kuramsal makine Neumann'ın fikirlerinin başarılı bir biçimde uygulanmasını içermektedir. (Gardner M, 1970)

Algoritmalar mimarlık disiplini için değil de planlama için geliştirilmiş ve uygulamaya konulmuştur. Bunun sebebi, birçok bilimsel çalışmalardakinin aksine mimarlık disiplinde, sezgisel hatta bazen bilinçsizce olagelen tasarım süreci, matematiksel eşitliklerle kolaylıkla ifade edilebilir bir süreç değildir. Mimari bir yapı oluşturmak için kod yazılırken, bir dizi komut bir araya getirilirken, hem tasarım hem de tasarım sürecini formülize edilmelidir. Bu konuda istisnalar da vardır: Christopher Alexander'ın HIDECS3 (1963) programı algoritmik olarak karmaşık tasarım işlerini basit parçalara bölüp sonra en ‘uygun’ tasarımı oluşturabilmek için tekrar birleştirmeyi amaçlamıştır. Rucker, “Alexander'ın algoritması mimari tasarımı gereğinden fazla rasyonel ve yapısal bir korseye çevirmekten başka bir şey yapamamıştır” demektedir.(Rucker, 2006)

İşleme, matematiksel olarak ifade edilebilen her tür bilgi işlem sürecine verilen genel bir isimdir. Bu kavram basit akıl yürütmeden başlayarak, daha dar anlamda bir algoritma, protokol veya ağ topolojisi ile ifade edilebilen iyi tanımlanmış bir modeli takip eden sürece kadar genel bir anlama sahiptir.

Geleneksel Matematikten “İşleme”ye (Computation)

1980'lerin başında matematikçi Stephen Wolfram “*karmaşık sistemler*”i *hücresel otomat* (cellular automata) yoluyla araştırmaya başlamıştır. Wolfram, matematikteki formalizmin “*kompleks davranışlı sistemler*”i çözümlemekte yetersiz kaldığını söylemektedir. Kompleks davranışlı sistemlerin, matematik yerine “*işleme*” (computation) yoluyla çözmek gerektiğini düşünmektedir. Kompleks davranışlı sistemleri çözmek için ise karmaşık işlemler ve karmaşık formül düzenleri kullanmanın karşısında durmaktadır. Doğadaki *karmaşık davranışlı sistemlerin üretici basit kurallarla* çözümlenebileceğini söylemektedir. (Wolfram, 2006)

Wolfram, “Doğadaki birçok sistem yüksek düzeyde karmaşık davranış sergilemektedir. Bunun yanı sıra bazı sistemler ise basit davranış biçimi sergilemektedir. Bu basit davranışlı sistemler, tekdüze veya tekrara dayalı bir çözüm içermektedir... Algoritma yazılımı doğadaki sistemler gibidir. Bazıları basit davranışlı bazıları ise karmaşık davranışlıdır. Sanılanın aksine karmaşık davranış gösteren programlar çözümlenmesi zor formüllere dayalı değildir. Tam tersine çok basit kuralların adım adım *işlenmesi* ile çözümlenebilmektedirler. Ama bu basit kuralların işlenmesiyle sonuçta beklenmedik karmaşık sonuçlar üreyebilmektedir. Doğadaki karmaşık sistemlerinde çözümünde basit kurallar yatmaktadır” demektedir. (Wolfram, 2006)

Wolfram kompleks fenomenini araştırmanın yanı sıra “kendini organize eden sistemler” (self-organize systems) ile ilgili de araştırmalar yapmıştır. Karmaşıklık gibi “kendini organize eden sistemler de ürettikleri sonuçların karmaşıklığı çözümlenmenin zor ve karmaşık olacağı anlamına gelmemektedir. Çok basit kuralların adım adım uygulanması ortaya karmaşık sistemler üretmektedir. Sonuç olarak kendini organize eden karmaşık bir sistem altında basit kurallar silsilesi bulunduğu somutlaştırılabilir. Wolfram bu çözümleme yönetimine “yeni bir çeşit bilim” adını vermiştir. Wolfram klasik matematikten işlemlemeye doğru kaymayı önermektedir. Bu yolla basit ve *karmaşık* algımızın değişeceğini düşünmektedir. (Rocker, 2006)

Algoritmaların mimarlıkta kullanılması günümüzde tekrar gündeme gelmektedir. Sayısal teknolojilerin hayatımızdaki yerinin önemi artmaya devam ederken bir yandan mimari tasarım sürecinin içine giren bilgisayarların daha verimli kullanılması üzerine çalışılmaktadır. Henüz tasarım yetisine sahip bilgisayarlar yapmak, insan zekasına sahip robotlar yapabilmek kadar uzak olsa da elimizdeki büyük soyutlama ve işlemleme gücünü kullanmak için araştırmalar yapılmaya devam etmektedir. Kompleks ve kendini organize eden bir sistem olarak görebileğimiz insanın tasarım becerisi, bilgisayar teknolojileri günümüz sosyal hayatını kökten değiştirirken, tekrar gözden geçirilmektedir.

Mimarların tasarım süreci, çok katmanlı bir süreçtir. İçerisindeki insan ve fikir akışıyla tasarım bir “*olagelen*” davranış sergilemektedir. Tasarım sürecinin işleyişi sırasında çok sayıda kararın aynı anda birbiri ile iletişim halinde alınmasını gerektirmektedir. Tasarım süreci, kendini organize eden bir kompleks sistem olarak değerlendirilmektedir. Günümüzde bu sistemlerin bir genetik modül üzerinden basit kurallarla çözümlenmesi yoluna

gidilmektedir. Deleuze'ün soyut makine'si, Neumann'ın hücresel otomat'ı tasarım süreci gibi kompleks sistemleri algoritmik düşünce ile çözümlenebileceğini belirtmektedir.

3.2 Tasarım Nedir?

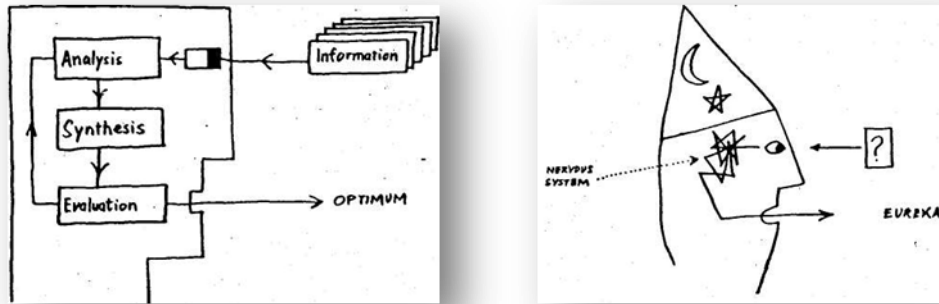
Tasarım insana verilmiş en büyük yetenektir. Her insan tasarımcıdır; tasarımı istendiği yönde eğip bükebilir. Tasarım etimolojik olarak “Tasa-r-ı-m” köklerinden gelmektedir. *Tasa* kaygı, dert olarak açıklanabilir. Tasarım kavramı, bir dertlenme olarak değerlendirilebilir. *Tasar* ise “bir iş, bir düşünce sırasını, düzeyini gösteren resim, yazı, plan” olarak belirtilmektedir. *Tasarı* ise “olması veya yapılması istenen bir şeyin zihinde aldığı biçim, proje”dir. “Bir araştırma sürecinin çeşitli dönemlerinde izlenecek yol ve işlemleri tasarlayan çerçeve”ye ise *tasarım* denilmektedir. [1] **Tasarım**; yapılacak eylem için izlenecek yolun, düşünce sırasının biçimlendirme kaygısıdır, denebilir. Tasarım kavramının, algoritma kavramına olan etimolojik yakınlığı dikkat çekicidir.

Araştırmacı ve teorisyenler, tasarım yeteneğinin, sistematüğini çözme çabasındadırlar. Tasarım üzerinde yapılan yöntem çalışmalarının tasarımı açıklayabilmesi çeşitli araştırmalar yapılmıştır. “Ancak, giderek tasarıma bilim açısından yaklaşma gerekliliği duyulmaya başlanmıştır. Şöyle ki, tasarımın büyük ölçüde insan beyninde oluşan bir süreçle ortaya çıkması nedeniyle, insanın psikolojik ve toplumsal açıdan dünyaya bakışı, çevresini etkileyişi ve etkilenişi tasarıma yansır. Bu nedenle, insanın tasarlama yaparkenki davranışları tasarlama metotlarını tanımlamada ve ortaya koymada önem kazanmaktadır. Bu da ancak, tasarımcı üzerinde yapılacak bilimsel çalışmalarla sağlanabilmektedir.” (Bayazıd, 2004) Sonuçları da tasarımcıdan tasarımcıya değişmektedir.

Terzidis, tasarım teriminin genellikle planlama terimiyle karıştırıldığı söylemektedir. Planlama, önceden belirlenen bir amacı yerine getirebilmek için yapılan şema, program ya da metodu ortaya koyma çabasıdır. Tasarım ise kavramsal ve soyut bir aktivitedir; bir *fikri* formüle etmek, görünür bir form veya aktiviteye dönüştürme niyetidir. Tasarım kavramsallaştırma, kurgulama, hayal etme, yorumlamadır. Planlama ise gerçekleştirme, organizasyon, uygulamadır. Tasarım belirsiz, tartışmaya açık bir oluşumdur. Tasarım, *olagelir* ya da başlangıç noktası genellikle kesin olmayan bir uygulamanın zihinde belirmesidir. Tasarım planlamanın çerçevesini çizen ilk aşamadır. Tasarım her zaman planlamadan önce gelir. (Terzidis, 2006)

J. Christopher Jones, sistemli hale getirilmemiş tasarım süreçlerinin yarattığı tatminsizlikler ve yüksek maliyetli tasarım hataları sebebiyle özellikle tasarım süreçlerinin dışa vurulması gerektiğini belirtmektedir. Bu yolla her çizgisi yüksek maliyetlerle hayata geçirilen mimar ve mühendislerin daha az hata yapacaklarını söylemektedir. Jones'un tanımladığı iki tür tasarımcı tipi vardır. Bunlar “kara kutu” yani “sihirbaz tasarımcı” ve “cam kutu” yani “bilgisayar tasarımcı”dır. (Şekil 3.7) Kara kutu tasarımcılar, tasarım sürecinin kesinlikle irdelenemeyeceğini tamamıyla içsel bir işlem olduğunu savunmaktadırlar. Tasarımın psikolojik bir olgu olduğunu ve anlık sıçramalarla yapıldığını ya da ilham geldiğini söylemektedirler. Tasarımcıyı hızlandıranın ise tasarımcı deneyimi olduğunu savunmaktadırlar. Yaratıcılığın dışa vurulmasının mümkün olmadığını dışa vurulmaya çalışılan tasarımın da yaratıcılıktan uzak sıradan ve basit ürünler vereceğini savunmaktadırlar. (Jones, 1969)

Cam kutu tasarımcılar ise bir bilgisayar gibi çalışmakta ve mistik çözümler yerine rasyonel çözümleri kullanmaktadırlar. Bu tasarımcılar tasarım sürecinin tamamıyla açıklanabilecek bir süreç olduğunu belirtmektedirler. Sistematik tasarım süreçlerini kullanan tasarımcılar, bu süreçte, neyi ne için yaptıkları bildiklerini vurgularlar. Fakat sonuç olarak cam kutu tasarımcılar bazı tasarım süreçlerinde kara kutu sürecinden daha iyi sonuçlar elde etse de bazı durumlarda cam kutu tasarımcılar da kara kutu tasarımcısı davranışlarını sergilemektedirler. (Jones, 1969)



Şekil 3.7 Sihirbaz Tasarımcı - Bilgisayar Tasarımcı

Bu iki yönteminde farklı problemlerde farklı avantajları bulunabilmektedir. Bir mühendislik tasarımı probleminde en önemli girdi her zaman verimdir. Bu noktada cam kutu oldukça yararlı bir yöntem olabilmektedir. Örneğin özellikli bir mimari problemin en önemli girdisi

verim olmayabilir. Verimin bazı estetik ve psikolojik girdilerden sonra gelmesi genellikle mimari tasarım süreçlerinde görülmektedir. Sonuç olarak, kara kutu, kimi zaman cam kutuya göre beklenmedik derecede etkileyici çözümler sunabilmektedir. Jones ve Alexander gibi birçok araştırmacı, tasarımın haritasını sunmak için birçok yöntem ortaya atmıştır ve bazı açılardan tasarımcılara yardımcı olmuşlardır. Yine de yapılan çalışmalar tasarımcıyı, tek tip veya birkaç tip ile sınırlamaya çalışmaktadır. Her tasarımcı için özdeş bir kalıp uydurma çabası doğru sonuçlar vermekten uzaktır. Tasarımcıların farklı ürünler sunmasının asıl sebebi tek tip olmamalarıdır. Tasarımcılar tanımlanırken diğer bir tasarımcı da olmayan özellikleriyle tanımlanmaktadır. Bir tasarımcı için bu tasarım cam kutudur veya kara kutudur demek doğru sonuçlar vermeyecektir.

Tasarım fikrinin altyapısı Antik Yunan'dan günümüze kadar tartışmalara açık bir konu olmuştur. Plato'ya göre onun dünyasında her şey idealdir. Dünyada bizim gördüklerimiz onların sadece birer yansımasıdır. Bu açıdan bakıldığında tasarım süreci cam bir kutu olabilir. Deleuze göre varlık, o varlığın ilişkilerinin bütünüdür. Onun dünyasında her varlık çevresiyle ilişkilerinden vardır. İlişkisiz bir varlık yoktur. Bir örümcek ağı düşünüldüğünde her bir varlığı, üç boyutlu bir örümcek ağı (Şekil 3.8) içerisinde iplerin düğüm noktası olarak açıklanabilir. Varlık bir ilişkiler düğümüdür. Düğümlenecek ilişki yoksa düğüm de yoktur. Belirtilen düğüm Deleuze tarafından çokluk (*multiplicity*) olarak tanımlanmaktadır. (Raychman, 2001) Hegel, “bir şeyin “olduğunu”, ama bu “olma”sı dışında hiç bir nitelik ya da karakteristiği olmadığını söylemek, bu şeyin hiçbir şey olmadığını söylemekle birdir.” demiştir.



Şekil 3.8 Örümcek Ağı

İnsanı oluşturan fiziksel varlığı değil; tecrübe ettiği ve yaşadığı her şey ile ilişkilenebilir. Bu açıdan baktığımızda, hiçbir zaman tasarımcının bir cam kutu olması beklenemez. Çünkü tasarımcı her zaman birebir açıklayamayacağı tasarıma etki eden unsurlar olacaktır. Tasarımcının sürecini açıklamak için tek bir yöntem yeterli olmamaktadır. Hiçbir tasarımcı tamamıyla cam veya kara kutu olamamaktadır. Her tasarımcı gri kutulardır ve renkleri grinin tonlarıdır, denebilir. Tasarımcı kendi rengini bilmeli ve problem el verdiğince, süreci şeffaflaştırmaya çalışmalıdır. Şu anda tasarım süreci, formüle edilmekten uzaktır. Goldenberg, “Tasarımcı yaratacağı ürünün sesini dinlemeli” demektedir. (Goldenberg 2003) Her tasarım için tasarım süreci diğer bir süreçten farklı olmaktadır. Bunun yanı sıra her tasarımcı da bir diğer tasarımcıdan farklıdır. Bu sebeple tasarımcı ürünü tasarlamadan önce tasarım sürecini tasarlamalıdır.

Tasarımcının “ilk eskizi” yapmasına Deleuze yaklaşımıyla bakarsak; bu eskiz tasarımcının bir saniye önce gördüğü bir fotoğrafla veya yıllar öncesi yaşadığı herhangi bir olayla ilişkide olabilir. Ayrıca tasarımcı hangi ilişkilerin düğüm noktasında, nasıl bir motivasyonla, bir veya daha fazla anlık sıçramayla, bu noktaya gelmiştir? Bu soruları elimizdeki imkanlarla cevaplanamamaktadır. Terzidis’e tasarım ve planlama süreçlerinin birbirlerine karıştırıldığını söylemektedir. Bunun sebebi planlama ve tasarım birbirine kaynaşmış ve ayrıştırılmaz süreçler olmasıdır. Ürünün fonksiyonu, ürünün değeri için ne kadar önemliyse ürün için planlama süreci o kadar ön plana çıkmaktadır. Sanatsal bir tasarımla mimari tasarım arasındaki en önemli fark mimari tasarımda, fonksiyon asla devre dışı bırakılamamaktadır. Mimarlık disiplinin ilk amacı *barınma* ihtiyacını karşılamaktır. Tasarım ve planlama süreci mimari süreç içerisinde birbirlerine kaynaşmışlardır. Mimar, gerekirse fonksiyonu da yorumlayabilmekte ve tasarımın ilk fikir sürecine girdi olarak kullanabilmektedir. Goldenberg’in dediği gibi ürünün sesini dinlemek gerekmektedir. Her ürün ve her tasarım süreci, belirli ortak paydalara sahip olabilsen de (uygun:fit:özgül) tasarım ürününü özgünleştiren (fark:different:populasyon) ortak payda dışında kalan fikirlerin birbirleriyle kurdukları (ilişki:relationship:topolojik) ilişkiler ağıdır.

Tasarım süreci, bilim alanında adı geçen kompleks sistemler içinde değerlendirilmektedir. Kompleks sistemler çözümlenmesi imkansız olduğu varsayılan sistemlerdir. Kompleks sistemler girdileri ve bu girdilerin ortaya koyacağı sonuçların belirsizliğiyle tasarım kavramıyla örtüşmektedir. Bu soruna günümüzde uygun cevapların verilememesi tasarım

sürecini, şeffaflaştırılma denemelerinden vazgeçilmesi anlamına gelmemektedir. Kompleks sistemlerin çözümlenmesi üzerine yapılan bilimsel çalışmalar sonucunda, bu sistemlerin çözümüne en uygun yöntemin *algoritmik düşünce sistemi* olduğu belirtilmektedir.

Tasarım sürecini tasarlamak üzerine yapılan çalışmalar, tartışmaya açık kurgusal sonuçlar verebilmektedir. Bilgisayar ve iletişim teknolojilerinin, sosyal ve kültürel hayatı küresel ölçekte etkilediği açıktır. Bu değişimle birlikte Alexander'dan sonra algoritmik düşünce tekrar gündeme gelmiştir. Tasarım sürecini tasarlamak için bir yöntem olarak kullanılabilir. Algoritmik düşünce bir fikir soyutlamasıdır.

3.3 Tasarım ve Algoritma

Temsil aracı olarak kullanılan sayısal görsel medya, artık mimarlığın bin yıllardır üzerine düşündüğü tasarım sürecine yardımcı olması için kullanılmaktadır. Dijitalleşen mimari temsil, mimari tasarımın sürecine de etkide bulunmaktadır. Bazı tasarımcılar son zamanlarda yeni form arayışlarına kökten bir değişim getirmeye çalışmaktadır. Temsil dışı bilgisayar kullanımı ya da başka bir deyişle “tasarım destek aracı” olarak bilgisayar kullanımı mimari tasarım içine girdiği noktada kendini türeten geometrik algoritmaların da tartışılması gerekir. Bu algoritmalar yeni geometri kavramları getirmektedir.

Bilgisayar ortamında tasarım fikri, bazı geometrik kavramların üzerine durulmasını gerektirmektedir. Çok eğrilikli yüzeylerin temsil sırasında rahatlıkla kullanılması CNC makineleriyle BDÜ ile üretiminin kolaylaşması, bu formların mimari tasarımın içine öncelikle biçimsel olarak daha sonra fikir tabanıyla, araştırmalara girmesini sağlamıştır. Deleuze'ün geometriden aldığı topolojik düşünce ve Mobius bandı gibi (Şekil 2.2) kavramlarını analitik ve algoritmik olarak bilgisayara tanıtmak amacıyla ortaya çıkan NURBS kavramı mimarların Öklid dışı geometrileri kullanmasını kolaylaştırmaktadır. Non-Uniform Rational B-Splines kelimelerinin baş harflerine karşılık gelen NURBS, çok eğrilikli bir yüzeyi tanımlamakta kullanılmaktadır. NURBS'ler eğri çizgiler plastik, ahşap veya metalden yapılmış esnek bant şeritlerdir. Öklid geometrilerinde olduğu gibi yüzeylerden katı oluşturmak yerine üç boyutlu katıyı oluşturmak için NURBS'ler kullanılabilir. NURBS'ler mimari tasarıma girdikçe bu geometrileri fonksiyonel olarak tanımlamak ve parametrikleştirmek gerekmektedir. Kolarevic parametriklerin mimari tasarımda kullanımı için; “Parametrikler ise mimari tasarımda sabit ve değişebilen, tekil ve çoğul işlemlerinin yerini değiştirerek yapar. Tasarımcı üreteceği mekana

dair deęiřken bilgileri sabitlere baęlı olarak deęiřtirerek tasarım yapılabilmektedir” demektedir. (Kolarevic 2003)

Algoritmalar, tek bir sonucu olan matematik denklemlerin tekrarıdır. Parametreleri sabit algoritmanın sonucu tektir. Parametreleri devamlı deęiřtirerek veya tasarım konsepti iinde belli deęerler arasında bilgisayarın geliři gzel semesine izin verilerek ortaya aynı tektonikte sayısız rn elde edilebilir. Sonu rnlerin tektonięi algoritmanın kendisi olmaktadır.

Algoritma (Script) yazma iřinin hayranlık uyandıran parası, tasarım abasına ek bir katmanın nasıl eklemlendięidir. Tasarım srecine olan bakıř aısının deęiřimine gerek duyulmaktadır. Bir yazılım bu srece eklemlenebilir ve bu srece ivme kazandırabilir. Algoritma mantıęının teorik altyapısı hiyerarřık, kapalı, tmden gelen bir sistematikten ok aık ve tme varan bir morfoloji grnmne sahiptir. Sonu olarak bu hareket sadece gnmzde moda olmaktan ziyade tasarım sistematıęine kkten eklemlenmeler ve biimlenme deęiřimleri getirebilir. (McCullough 2006)

Algoritmanın tasarım srecinde kullanılması eřitli avantajları da yanında getirmektedir. Dijital tasarım sayesinde dijital fabrikasyon srecine daha hakim olunabilmektedir. Formun ifade ediliř geleneklerinin, teorik temelleri daha kolay algılanabilmektedir. Algoritmik beęeni ya da biyomorfik biimlenme, greceli olarak daha rahat ortaya ıkmaktadır. zellikle byme ve ortaya ıkma yanı sıra harmoni ve z yinleme, kendinden treme ana fikirleri mimarlıęın kodlama ihtiyacını arttırmaktadır. Algoritmik kodlama mřteri isteęiyle belirli parametreler iinde kalarak deęiřimi de kolaylařtırmaktadır. (McCullough 2006)

Script dilleri kullanarak, gnmz  boyutlu programların, retici tarafından belirlenen kuralların dıřına ıkılabilmektedir. Terzidis ve Chu arařtırmalarını basit, aıka belirtilmiř kurallar ierisinde nceden kestirilemez karmařık davranıřlı sistemler elde etmek iin kullanmaktadırlar. (Rocker 2006)

İřleme, sayısallařma ve algoritmik sre st dzeyde bir soyutlama biimidir. Algoritmik strktrler algılanan rn ile baęlı olma zorunluluęu olmaya soyut paternler ierirler. Algoritmik yazılımlar maddeleřmemiř bir aratır daha ok bir dřnce gibidir.

Son olarak alıřma alanını kiřiselleřtiren ve kodlamayla zihinsel haz alan ve farklılařan bilgisayar ile bymř yeni bir tasarımcı kitlesi gelmektedir. řu zamanda grsel ara yz daha ekici olan programlar gnmz insanı iin algı seviyesindeyken gn geldięinde

kodlama yaptığıının farkına varmadan yazdığı koda bakarken paternler gören ve bu paternlerin strüktürel yapısı içinde özgün tasarım ilkeleri üretebilen yeni nesil tasarımcılar ortada olacaktır. Günümüzde nasıl kağıt üzerindeki çizgiler mimari kodlama sistematığının bel kemiğiye gelecekte algoritmik kodlama biçimde algılanabilecek kavram tasarımlar ortaya çıkacaktır. Bu kesinlikle yeni bir idrak biçimi olacaktır. (Mccullough 2006)

Algoritmik düşünce sisteminin kendine has bir semantiği vardır. Mimarlık disiplininde de fikir soyutlamaları ve doğal dillerden farklı kullandığı bir semantiği vardır. Mimarlar tasarım, çizim ve sunum aşamalarında soyutlamayı fazlasıyla kullanır. Bu soyutlama, mimarlığın kendini kodlamasıdır.

Mimari Kodlama

Mimarlıkta “kodlama”nın anlamını ve rolünü anlayabilmek için kodun mimarlıktaki rolü ve geçmişini gözden geçirmek gerekmektedir. Kod, Latince “codex” teriminden gelmektedir. Roma imparatorluğu döneminde ahşap tabletlerin şekillendirilmesiyle elde edilen dokümanlara verilen isimdir. Bu “codex”ler sistem prensiplerini ve kurallarını içermekteydi. Mimarlıkta ise “prensip ve kurallar” mimarlık tarihi boyunca tasarımın baskın öncelikleri olmuştur. Diğer mimari üretim basamaklarının geçmişine bakıldığında mimarlık daima kodların değişimi ve içeriğiyle sınırlandırılmış ve şekillendirilmiştir. Sadece mimarlık kavramı değil onun medyası da bir çeşit kodlamadır. Bir dil, bir semantiktir. Kurşun kalem çizimlerinden makete, perspektiflerden teknik uygulama çizimlerine kadar mimarlık daima “sistemin prensiplerini ve kurallarını” belirtmek için kodlanmıştır. Mimarlıktaki kodlama teknikleri bilgisayarın mimarların hayatına girmesiyle çok daha ön plana çıkmıştır. Mimarlık tasarım sürecinin en başından sonuna kadar kendine has bir kodlama kullanmaktadır. Hatta mimarlık eğitiminin amacı öğrencilerin bu kodlamayı anlaması ve kendi fikirlerini en iyi biçimde bu kodlama biçimini kullanarak ortaya dökmesi üzerinedir. Rocker, “Konseptten projenin sunumuna kadar bir fikir kodlaması söz konusudur” demektedir. (Rocker, 2006)

3.3.1 Tasarımcı ile Bilgisayar Kodlaması

Farklı yazılım ve programların kullanımı artık mimari tasarım teori ve tartışma platformunun önemli bir elemanı haline gelmiştir. (Williams, 2004) Bu yüzden mimarların bilgisayarla iletişimini doğru yönlenebilmesi için bir bilgisayar ile doğru iletişimi kurması gerekmektedir. Bilgisayar evlerimizde ofislerimizdeki içi devrelerle doldurulmuş bir kutuya bağlı ekran ve

çevre birimlerinden çok, bir organizasyon ve dillenme biçimi olarak değerlendirilmelidir. Bu fiziksel hesap makinesinin altında, Deleuze'un "soyut makinesi" gibi bir soyut "evrensel makine" yatmaktadır. Bu evrensel makine tamamen kişisel bir ürün halinde kullanılabilir. Kişinin sosyal, kültürel, mesleki durumuna göre şekillenerek kişiye özel bir makine olmaktadır. Bilgisayar ile kurulacak doğru iletişimle bu evrensel makineler kişiye çok daha yararlı olmaktadır. Doğru iletişimi kurmak için bilgisayarın dili hakkında bilgi sahibi olmak gerekir.

Bilgisayarlar ile iletişime geçmek için ilk zamanlar makine dili kullanılmaktaydı. Makine dili karmaşık ve anlaşılması uzmanlık isteyen bir konudur. Makine dili, mikroişlemci ya da mikro denetleyici gibi komut işleme yeteneğine sahip entegrelerin, işleyebilecekleri komutlardan ve buna uygun sözdiziminden oluşan dile verilen addır. Makine dili, işlemcinin verilen komutlar doğrultusunda çalıştırılmasını sağlayan ve işlemci mimarisine göre değişen en alt seviyedeki (*low level*) programlama dilidir. Bu dil sadece 0 ve 1 ikililerinin (ikilik sayı tabanı) anlamlı kombinasyonlarından meydana gelmektedir. Bu dil temelde Jacquard'ın dokuma tezgahında kullandığı delikli kartlardan farklı değildir. 1'ler delik 0'lar delik olmayan anlamına gelmektedir. Bu nedenle, makine dilinin anlaşılması çok güçtür. 0 ve 1 ikilileri işlemcinin talimatlar seti doğrultusunda, uygulanacak operasyon, operasyonun gerçekleştirileceği verinin hafızada bulunduğu adres ve hafızaya ulaşım yolları gibi bilgileri ifade edecek şekilde bir araya gelmekte ve işlemci tarafından deşifre edilerek gerekli işlemin yerine getirilmesi sağlanmaktadır. Bu dil, donanımı doğrudan kontrol etme gücü olduğu için kullanılır. Bir bilgisayar yalnızca kendi makine dilini doğrudan anlayabilir. Makine dili bilgisayarın doğal dilidir ve bilgisayarın donanımsal tasarımına bağlıdır.[2]

Kullanılan işlemcinin komut setinden ibaret olan makine dili, komutları donanıma bağımlıdır. Günümüzde kullanılan i386 (32bit - Intel) ve i486 gibi işlemci standartlarının her birine ait birer komut seti bulunmaktadır ve bu komut seti yalnız o mimariye yöneliktir.[2]

Örneğin: 1 0011 0111 gibi 9 bitlik bir bütünü düşünecek olursak burada sol baştaki 1 biti (*BIT* *bilişimde, kullanılan en küçük bilgi birimi. Evet/Hayır, 0/1, açık/kapalı, var/yok, doğru/yanlış gibi düşünülebilecek birbirine karşıt iki durumu belirtmekte kullanılır. Kök olarak ikili basamak (veya başka bir görüşe göre ikili birim) anlamlarına gelen 'binary digit' (veya 'binary unit') kelimelerinin kısaltılmasından gelir*) dolaylı adresleme yapıldığını, bu bitten sonraki 4 bit olan 0011 çarpma işleminin yapılması gerektiğini geriye kalan 4 bit 0111 ise

hafızanın 0111 adresine gidilmesi gerektiğini belirtebilmektedir. Bu düzen tabi ki farklı işlemci mimarileri için farklı olacaktır.[2]

Grafik tasarımcısı, bilgisayar programcısı ve üniversite profesörü olan Casey Reas: “*Bir program çekirdeğinde kesin talimatlar setidir ve bilgisayara kusursuz olarak ne yapması gerektiğini söyler, bir makine tarafından anlaşılabilir bir dizi kelime ve sembol ile anlatılan fikir kodları iskeletidir. Her program dili bir kelime ve sembol derlemesidir (sözdizimi (sentaks))ve bir programın kullanacağı kurallar dizisi tanımlar (semantik). Her farklı programlama dili insanların fikirlerini kodlara çevirmeye yarayan farklı birer yoldur*” demektedir. (Reas, 2004) Programlama dilleri insanlar ile makineler arasında bir ara yüzdür. Her programlama dili temelde, makine kodunun daha kolay bir biçimde insanlar tarafından yazılması ve takip edilmesi amacıyla üretilmiştir. Tarih içinde ilk olarak günümüzde *alçak düzey dil (low-level)* olarak isimlendirilen *Assembly* dili ortaya konmuştur. Zamanla programlama dilleri makine dilinden insanın daha kolay anlayabileceği bir dile doğru değiştikçe diller, *orta düzey(middle level)*, *üst düzey (high level)*ve son olarak *en üst düzey veya görsel dillere (visual)* doğru isimlendirilmiştir. Farklı düzeylerde programlama dilleri, farklı disiplinlerin ihtiyaçlarına yönelik disiplinlerin ihtiyaçları üzerinden üretilmiştir.

Tasarımcılar günümüzde farklı disiplinlerin ihtiyaçlarından doğmuş olan *sözdizimleri (sentaks)* ile uğraşmaktadır. Kullanılan bilgisayar dilleri tasarımcı için limitli imkanlara sahiptir. Her tasarımcı kendisinin bilgisayarla doğrudan ilişkiye girmek için kendi dilini yazma ihtimali ortadadır. Bu bilgisayar programlama dilleriyle ilgili hiçbir pratiği olmayan tasarımcılar için can sıkıcı olabilmektedir. Bu durumun yarattığı eksiklikle ortaya son dönemde tasarımcılara CAD tabanlı programların içeriğinde makro dili (script) bulundurulmaktadır. Makro dilleri, sistem bütünleştirici diller (System Integration Languages) olarak da bilinmektedir. Uygulamaları hızla birleştirip birbirine bağlayan ve bunlar arasında iletişimi sağlamakta kullanılmaktadır. Makro dillerinin (script) sözdizimi ve semantiği, programcı olmayan kişiler tarafından daha kolay anlaşılabilir geliştirilebilir. Bu diller tasarımcı tarafından bilgisayar programlarının tasarım sürecinde kullanılabilir yararlı bir yol olarak modifiye edilmesini sağlamaktadır. Bu sayede bilgisayar teknolojilerinin tasarım açısından limitlerini aranmakta ve açık bir biçimde tasarım mantığını genişletilmekte, tasarım sınırları daha geniş bir alana yayarak yeni ihtimaller üzerine çalışılmaktadır. Makro dillerinin kullanımı ve farklı tasarım ihtiyaçları için özelleştirmeye verilebilecek en popüler örnek,

Frank Gehry ve Gehry Ortakları'nın Dassault Sistemleri'nin havacılık ve uzay endüstrileri için ürettiği CATIA programının gerekli eklemelerle ile yapım teknolojileri için bir tasarım aracı haline getirilmesidir.

Makro dilleri, müzik notaları, dans koreografisi, sinema veya tiyatro senaryoları gibi görülebilir. Performans gösterecek kişilerin ne yapması gerektiğini talimatlar ve yorumlarla anlatmaktadır. Makro dilleri aslında bütünü oluşturan parçaların kompozisyonun talimatlarını verir.

Bilgisayar kodu veya başka bir deyişle algoritması aynı mimarlık kodlaması gibi talimatlar seti içeren bir dildir. Kodlama, yüksek düzeyde bir soyutlama biçimidir. Mimarlar ve mimarlık araştırmacıları son yıllarda bu soyutlama biçimine oldukça ilgi göstermektedir. Mimarlık ve onun ürünleri bir fikrin soyutlaması ve onun dillendirilmesi biçimdedir. Günümüzde amaçlanan, bilgisayarın mimarları kırtasiye işlerinden kurtarmasından çok, bu iki soyutlama biçimini üst üste gelecek şekilde karşılaştırmayı başarmak üzerinedir. Sonuçta bilgisayarlar artık sadece mimari sunumu desteklemekten öte mimarlık konseptinin de tekrardan ele alınarak bu konseptlerin yeniden dillendirme çabası halini almaktadır.

Mimari tasarım sürecini kompleks bir sistem olarak değerlendirebiliriz. Bu süreci somutlaştırmak ise oldukça zordur. Bunun sebebi her tasarımcının kendine ait bir bilgi birikimi ve tasarım felsefesinin bulunması olarak görülebilir. Tasarım sürecinin kişiden kişiye değişebilecek somutlaştırma yöntemleri olabileceği için süreç için ortaya konacak genellemeler doğru sonuçlar vermekten uzaktır fakat bilgisayarın kişileştirilebilir arayüzleriyle birlikte, tasarımcıya yeni ufuklar açılabilmektedir.

Algoritmik mimarlık, bilgisayarların mimari tasarım ve üretim süreçlerindeki potansiyelinin araştırılması ve daha iyi anlaşılması için yeni bir kapı, çalışmak için yeni bir alan açmaktadır.

3.3.2 Algoritmik Mimari Tasarım

Algoritmik düşüncenin mimari tasarımda özellikle kullanıldığı çeşitli tasarım yöntemleri günümüzde sınıflandırılmaya çalışılmaktadır. Bu kavramların fikir yapısı aynıdır ve hepsi de **parametrik** girdilerin değerlendirildiği **evrim** analogisinin kurulduğu **türetici** tasarım biçimleridir. Algoritmanın dillendirilmesi olarak yazılım, tasarım sürecinin genetik kodu olarak algılanmaktadır. Fikrin algoritmik adımlanması Deleuze'un özgül düşüncesiyle çıkmaktadır. Algoritma genotiptir. *Türettiği* ürünler ise farklı fenotiplerde bir popülasyon

oluşturmaktadır. Parametrik *farklar* Deleuze’ün popülasyon düşüncesinde var olan “üretken farklar”dır. Bu fenotip ürünler birbirleriyle topolojik olarak *ilişkidedir*.

Algoritmik mimari tasarım içeriğinde genetik kavramıyla anılmaktadır. Genetik kavramı kendini organize eden ve kendini üreten kavramlarını beraberinde getirmektedir. Kendini üretme ya da kendini kopyalama kavramları türetici tasarımı işaret eder. Algoritmik tasarım, türetici tasarım, parametrik tasarım, evrimsel tasarım, genetik tasarım günümüz mimari literatüründe farklı tasarım biçimleriymiş gibi açıklansa da temelleri biyolojik büyümenin algoritmik olarak soyutlanması mantığıyla çakışmaktadır. Temelde *özüml* bir modülün belirli *ilişki* kurallarıyla kendini kopyalayarak tekrar etmesiyle, ihtiyaç için *uygun* fakat birbirlerinden *farklı* ürünlerin elde edilmesidir.

Tasarım algoritmik olarak tanımlanması fikri beraberinde, parametrikleşme, türeticilik ve evrimsel olma kavramlarını getirmektedir.

Parametrik Tasarım

Akipek parametrik tasarım için: “Parametrik tasarım, tasarımın belirlenen parametreler üzerine kurulmasıyla ilgilidir. Mimari tasarım sürecinde parametrik tasarımın kullanıldığı örneklerde, rüzgar şiddeti, deniz tuzluluk oranı, su miktarı, insan akışındaki yoğunluk gibi çevresel veriler tasarım sürecinde parametreler olarak tanımlanmaktadır. Bilgisayar ortamında kurgulanan sistemde, parametrelere girilen farklı değerler sonucu oluşan değişim, tasarım aşamasında form üretimi için ya da fiziksel mekanda ışık — ses — biçim değişimleri için kullanılır. Parametrik tasarım detay çözümleri ve strüktür tasarımı için de kullanılmaktadır. Tek bir prensip formül oluşturulur; ölçü, açı, kalınlık değişimlerinin gerektiği yerlerde, parametrelerin değerleri değiştirilir ve tek bir prensip detay çözümüne dayalı çeşitli çözümler oluşturulabilir” demektedir. (Akipek, 2004)

Dış etmenler, mimari tasarımda her zaman ilk sırada düşünülen girdilerdir. Daha öncede açıklandığı gibi parametrelerin varlığı olmadan algoritma kurmak mümkün değildir. Ayrıca parametrelerin varlığı daima bir algoritmayı işaret etmektedir. Bu iki kavram birbirleriyle iç içe girmiştir.

Türetici Tasarım

Türetme kelime anlamı olarak, bilinen bazı şeylerden yararlanılarak düşünce gücüyle yeni bir şey bulmak olarak tanımlanmaktadır [1]. Türetici tasarım bilgisayar programı algoritmalarıyla

ya da benzer matematiksel ya da mekanik süreçlerle oluşturulan tasarımıdır [2]. “Bilgisayar ortamında kurulan üretici tasarım sistemlerinde, belirli bir biçim grubundaki elemanların çeşitli kombinasyonlarının belirlenmesine dayalı alternatif çözümler, sistematik olarak araştırılır” (Akipek, 2004)

“Üretici tasarım modelleri dilbilim kökenli araştırmalara dayalı, gramer tabanlı bilgisayar destekli tasarım modelleridir. Gramerlere dayalı sistemlerde tasarım alternatiflerinin türetilmesi kullanıcının belirlediği kurallar setine bağlıdır. Bu kurallar setine göre türetilebilecek tüm olası alternatifler ve biçimsel kombinasyonlar bilgisayar tarafından türetilir ve tasarımcı geleneksel temsil ortamında araştıramayacağı sayıda alternatifi değerlendirme imkanına sahip olur” (Akipek, 2004)

Türetici tasarım algoritmik tasarımın üst başlığı olarak düşünülmektedir fakat üretici tasarım fikir alt yapısını algoritmik düşünce üzerine kurmuştur. Bu sebeple üretici tasarımı, algoritmik düşüncenin alt başlığı olarak düşünmek gerekmektedir.

Evrimsel Tasarım

Evrimsel tasarımın üzerinde mimari araştırmalarda, "emergent" – “olagelen” kavramıyla üzerinde durulmaktadır. Girilen parametrelere göre kendini değiştirebilen bu değişikliği de önceden verilmiş algoritma yoluyla yapabilen tasarım ürünüdür.

“Biçimsel öngörülere dayanmayan, biçim yerine performansa önem veren, adapte olma, kendisini yeniden düzenleme özelliklerine sahip, program kombinasyonlarına izin veren bir mimari yaklaşımla ilgili olarak kullanılmaktadır. Bu yaklaşımda mimarinin biçimi kullanıcı, program, bağlam girdilerine göre zaman içine evrimleşerek oluşmaktadır. Bu süreçte tasarımcının rolü belirli bir duyarlılıkla süreci hızlandırmak ve bu öngörüye ve analizlere dayalı bir sistem kurmaktır. Strüktür, tasarımdan ayrı bir sistem olarak algılanmaz; tasarımın başlangıcında strüktürle birlikte kurgulanan bir biçimlenişten bahsedilir” (Akipek, 2004).

Frazer, Evrimsel Mimari isimli kitabında bu süreci şöyle tarif eder: "Mimari konseptler bir dizi üretim kuralları olarak tanımlanır, bu konseptlerin evrimi, gelişimi sayısal olarak şifrelenebilir. Bir dizi üretim kuralı, çok sayıda prototip biçim türetebilir. Bu sonuç ürünler benzeşik bir çevredeki performanslarına bağlı olarak değerlendirilir. Sonuç genellikle beklenmediktir" (Kolarevic, 2003).

Genetik algoritma, evrimsel mimari yaklaşımında temel bir alt kavramdır. Doğada, canlıların

oluşumunda ve biçimlenmesinde yönlendirici olan genlerin işlevini bilgisayar ortamında algoritma ve kodlar üstlenir. Bilgisayar ortamındaki genetik algoritmalar, çoğalma, gen geçişi ve mutasyon kurallarının işlendiği kromozom sarmalı benzeri, kural dizili bir yapıya sahiptirler. Bu sarmala parametreler işlenir ve değerleri tasarım süreci boyunca sürekli değişir, Frazer'ın deyimiyle birbirine benzer biçimdeki "yapay organizmalar" oluşturulur ve bu organizmalar belirlenen "güçlülük" kriterlerine uyum gösterip göstermediklerine göre elenirler (Kolarevic, 2003).

Tasarım, girdilerinin fazla ve belirsizliği nedeniyle henüz bütünüyle algoritmik düşünce fikriyle örtüşmemektedir. Mimari tasarım sürecinde süreç kompleks bir sistem olarak değerlendirilebilir. Günümüz pozitif bilimleri kompleks sistemlerin, algoritmik birer sisteme sahip olduğunu belirtmektedir. Algoritmalar, kompleks sistemlerin “genetik bilgisi” olarak görülmektedir.

Yaşayan organizmaların DNA ile türetimini ve DNA'nın evrimi bir metafor olarak kullanıldığında organizmanın sadece bir DNA kodunun organizmanın formunun belirlediği ve onun büyümesinin bu anahtar koda bağlandığı olduğundan hareketle bir algoritmanın da bir obje veya mekana anahtar kod olabileceği tartışılmaktadır. Bu yolla bir yapının bir genetik kodu olabilecek ve bu yapı genişleme ve büyümelere açık olacaktır.

Genetik konsepti, kalıtsal özün kendine has bazı kurallarla tekrarı kendi üzerine katlanması, “pili”lenmesi mekanizmasını mantığa döker. Bu mantık “özü” prensibinin kodlanması, türetimin otonomisi, “işleme” (computation) mantığını tam ortasındadır. Mimarlık yeni teknik ve bilimsel gelişmelerle birlikte ütöpik yeni bir dünya üretimi çabasına girmiş olarak algılanmamalıdır. Gen ve genetik kelimeleri biyolojiyle açık bir bağ kurmaktan öte mimarlık için yeni bir konsept ve soyutlama biçimi sunmaktadır. Genetik mimarlık biyolojinin tekrar sunulması veya biyolojik formların taklidi olarak algılanmamalıdır. Bunu yerine John Von Neumann'ın çalışması olan hücresel otomasyonun (cellular automation) bir izi olarak görülebilir.

Modern hesaplama sistemleri ortaya konulurken Neumann “kendini tekrarlayan” fikrini ön plana çıkarmıştır. Bu fikir “bir makine kendini tekrar üretebilir ve kopyalayabilir” olarak algılanabilir. Kendini kopyalamak biyoloji biliminin en önemli kavramlarından biridir. Hücreler kendini kopyalayarak çoğalırlar. Biyolojinin her alanında bu kavram mevcuttur. Önemli olan

kopyalamanın prensibi, özüdür. Kopyalama hücrelerin genetik kodlarıyla yönlendirilmektedir. Mimarlık disiplini “*genetik*” kavramındaki biyolojik analogiyi bu noktada incelemektedir.

Bir ürünün genetik kodunun olması, o ürünün işlevine *uygun*, fakat birbirinden *farklı* fenotipte ürünlerin ortaya konmasını sağlamaktadır. Genetik fenomeni, mimarlık disiplinde algoritmik mimarlık ile ortaya çıkmaktadır.

Tasarım sürecinde algoritmaların kullanımı ile mimarlık disiplini, Deleuze’ün varlık felsefesinin fikir altyapısıyla ve biyolojideki evrim kavramıyla analogiler kurmaktadır. Günümüzde sürdürülebilir mimari olarak değerlendirilen bakış açısının ve doğaya uyumun önemini kavrandıkça algoritmik tasarımın önemi artacaktır. Bir organizma gibi doğaya uyum sağlayabilen, parametrelerini doğanın kendisinden alan, bu parametreler ışığında mimarın belirlediği genetik bir algoritma ile değişebilen ve kendini organize ederek gelişen kompleks mimari ürünlerden söz edilmektedir.

Üretimi ve temsili sayısallaşarak değişen mimari ürün üretme sürecinin erken aşaması olan tasarım süreci de bu sayısallaşma ile evrilmektedir. Bu değişimin getirdiği sonuçları daha iyi algılamak adına akademik düzlemlerde yapılan araştırma çalışmaları, mimari tasarımın geleceğine ilişkin ipuçları vermektedir.

4. TASARIMIN ALGORİTMİK DÜŞÜNCE İLE DENEYLENMESİ

Sözü edilen alana ait tüm kavramlar ve gelişmeler, tasarımın sürecini ve uygulamasını değiştirebilecek potansiyele sahiptir. Bu potansiyelin kullanımı tasarımcının hem tasarıma, hem de bilgisayara olan hakimiyetine bağlıdır. Tasarımcıların kendi tasarımlarının planlama algoritmalarını çıkararak (ya da bu konuda uzmanlarla çalışarak) bunu bilgisayar diline aktarabilmeleri gerekmektedir. Yaptığı tasarımı deşifre edebilen tasarımcılar, tasarımlarını basit ve temel kural ve elemanlara dökebilirler.

Tasarımın algoritmasının yazılmasında tasarıma ek bir katman gelmiş ve bu da tasarım sürecine üst düzeyde bir soyutlama eklenmesini sağlamıştır. Bu soyutlamada algoritmik kodlamayı gerçekleştirmek gereklidir. Bu algoritmik kodlamanın zaman içinde tasarım sürecinin temel bir parçası olacağı ve ürünün yerine algoritmanın tasarlanması gerekliliği artacaktır.

BDT programların içeriğinde makro dili (script) bulundurulmaktadır. Makro dilleri, daha önceki bölümlerde açıklandığı gibi, sistem bütünleştirici diller (System Integration Languages) olarak da bilinmektedir. Uygulamaları hızla birleştirip birbirine bağlayan ve bunlar arasında iletişimi sağlamakta kullanılmaktadır. Makro dillerinin (script) sözdizimi ve semantiği, programcı olmayan kişiler tarafından daha kolay anlaşılabilir geliştirilebilir.

Script dilleri, sistem programlama dillerinden (C++, Pascal, vb.) farklı olarak özel görevler için tasarlanmış yapılardır. Sistem programlama dilleri sıfırdan başlayıp veri tabanları ve tasarlanmış algoritmalar ile yeni sistemler üretirler. Script dilleri ise, mevcut bir dilin alt bütünlerini kullanarak, bu bütünleri bağlamayı hedefler; kompleks algoritma ve veri yapıları için kullanılmaz. Sistem programlama dillerinde sözdizimi kurallarına önem verilir. Script dillerinde ise uygulamayı hızlandırmak, tanımlanmış bütünleri bağlamayı kolaylaştırmak için sözdizimi kuralları sadeleştirilmiştir. Script dilinde elemanların nasıl kullanılacağını ve bir araya geleceğini gösteren tanımlanmış kalıplar yoktur. Herhangi bir çözüm için kullanılan alt bütün ve değerler tasarımcı tarafından öngörülmeyle başka bir durumdaki çözüm için kolaylıkla kullanılabilir (Ousterhout 1998).

Tasarımcıların bu semantik ve düşünce sistemine yaklaşımları çeşitlidir. Script dilinin ve tez kapsamında tartışılan kavramların, tasarım sürecine olan etkilerini deneylemek amaçlı bir atölye çalışması yapılmıştır.

4.1 Deneye Hazırlık

Belirlenen bir problem üzerine bir algoritma tasarlanarak bunun yazılımı (script) yazılmıştır. Tasarımcıların algoritmik düşünce ve mimari tasarımda yazılım dilleri konusundaki görüşlerini, yorumlarını almak için bir atölye çalışması düzenlenmiştir. Genel olarak günümüzde tasarımcıların bilgisayar diline hakim olmadıkları göz önüne alınarak, kullanıcı profili seçilmiştir. Tasarım sürecinde analitik ve algoritmik düşünceye karşı tepki gözlenmeye çalışılmıştır. Yazılım dili ve programı olarak da 3dsMax seçilmiştir. 3dsMax programı için yazılan “script”ler günümüzde algoritmaların bilgisayara dökülebilmelerini sağlayan araçlardan biridir. MaxScript’in uygulama alanı olarak seçilmesinin en önemli sebepleri arasında, yazılımın yaygın kullanımı, güçlü nesne yapısı ve yazılım hakkındaki mevcut kaynakların çokluğu gelmektedir. Bu üst düzey (high – level) bir yazılım dili olduğu için diğer makine dillerinden daha kolaydır. Ayrıca uygulamayı görselleştirerek ilk aşamada tasarımcılara kolaylıklar sağlamaktadır.

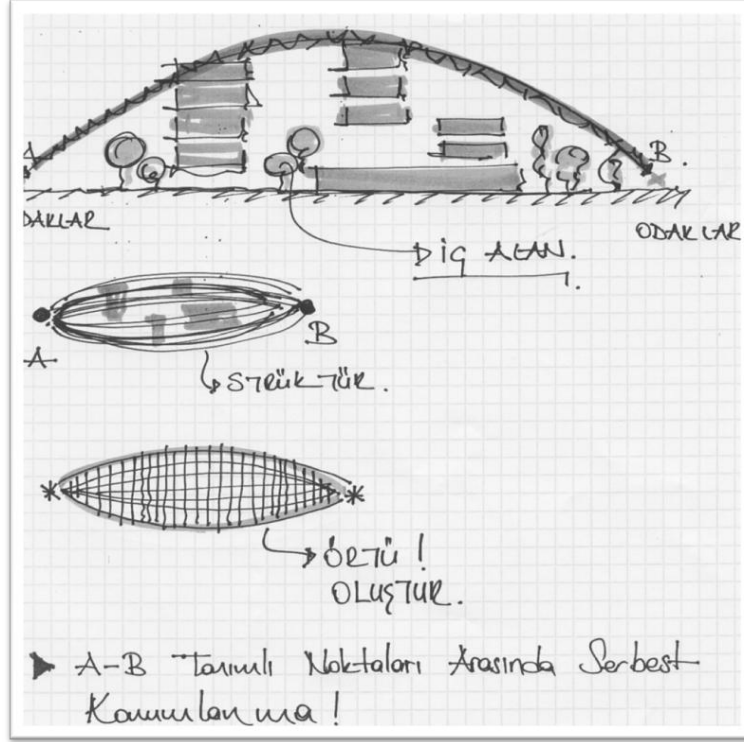
Alan çalışması için oluşturulan *problem* YTÜ Mimarlık Fakültesi Bilgisayar Ortamında Mimarlık Yüksek Lisans Programı kapsamında 2005 – 2006 güz yarıyılında uygulanan “Tasarımı Tasarlamak” atölyesi için oluşturulmuş problemi değiştirilerek oluşturulmuştur. Problem eskiz ile (en az) dört binanın yer aldığı kütle organizasyonu oluşturmak ve bunun üzerinin script yardımıyla örtülmesine yöneliktir.

Atölyenin *amacı* farklı profildeki mimarlar grubu ile ortaya atılan problemi çözmekten ziyade tez kapsamında tartışılan algoritmik düşüncenin mimarlar tarafından algısını test etmektir. Atölye çalışması, algoritmik düşünce ile geliştirilen scriptin farklı profildeki mimar kullanıcılar tarafından denenmesi üzerine olmuştur.

Atölye çalışması için kullanılan *yöntem* ise script yazımı ve bunun belirlenen kullanıcı profili ile denenmesi olmuştur. Deneyde öncelikle eskiz yoluyla kütle organizasyonu tasarımı ve ardından script yardımıyla organizasyonun üzerini örtecek biçimde bir üst örtünün tasarlanması ve son olarak yeni bir eskiz ile örtü tasarımının yorumu istenmiştir. Yazılan script daha önce uygulanan proje çalışması “Tasarımı Tasarlamak” atölyesi sırasında tez yazarı tarafından hazırlanmış scriptin geliştirilmesiyle elde edilmiştir. Script yazımı öncesinde fikrin algoritmasının ortaya çıkması gerekmektedir. Atölye çalışması sırasında kullanılacak scripti yazmadan önce, problemin analitik ve algoritmik olarak çözümlemek gerekmektedir.

4.1.1 Tasarım Probleminin Analitik ve Algoritmik Tabana Aktarılması

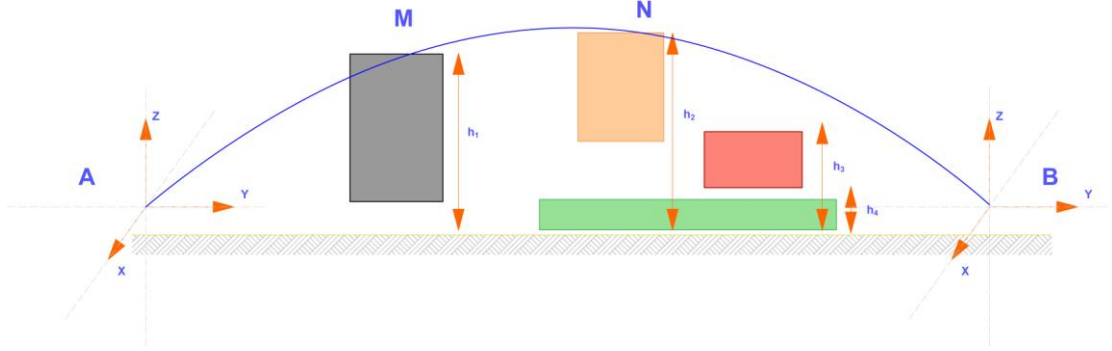
Bu probleme çözüm üretmeye yardımcı olabileceği düşünülen bir script yazımından hemen önce bir karakalem eskiz çalışması yapılmıştır (Şekil 4.1). Bu problem üzerine çalışacak bir script yazmak planlandığında, öncelikli olarak var olan eskizi matematiksel bir dille anlatmak gerekmektedir.



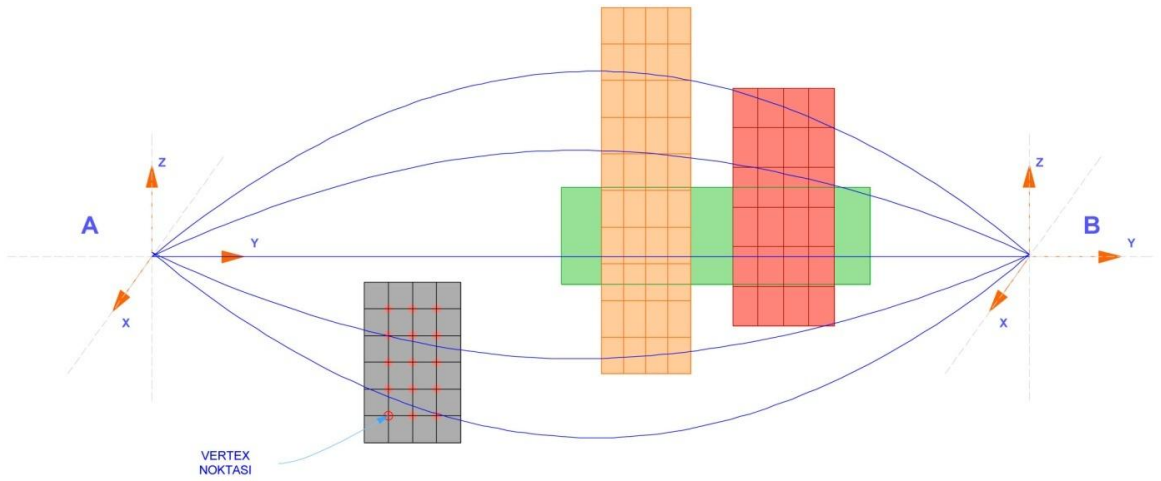
Şekil 4.1 İlk Eskiz

Eskizi algoritmik düşünce yoluyla tekrar düşülmesi için eskiz ya da diğer bir deyişle problemi çözmeye yönelik yaklaşımı, algoritmik olmalıdır. Bu amaçla eskiz, analitik olarak soyutlanmakta ve algoritma içinde kullanılması amacıyla parametrikleştirilmektedir. Kütleleri içine alacak ve örtecek biçimde çizilen üst örtünün iki ayrı odak noktası ve binaların yükseklikleri parametre olarak kullanılabilir. Bu eskizin altında yatan fikri sayısal olarak algılamak için, bu belirtilen parametreler analitik ortamda değerlendirilmelidir. Öncelikli amaç dört noktadan geçen bir NURBS çizmek olmalıdır. Noktalara A, M, N, B denebilir. Odak noktalarının analitik ifadesi olan (X,Y,Z) değerlerini belirlemek gerekmektedir. Üst örtünün başlangıç ve bitiş noktaları A ve B'yi analitik olarak belirlemenin ardından yapılması gereken üst örtünün yükseklik parametrelerinin belirlenmesi olmaktadır. Örtünün yüksekliği bina kütlelerinin üst yüzeyi olmalıdır. Üst örtü A(XYZ) noktasından başlayacak bina üst

yüzeyinden alınan herhangi bir $M(XYZ)$ noktasından ardından diğer bir kütlelerin üst yüzeyinin $N(XYZ)$ noktasından geçmeli ve $B(XYZ)$ noktasında sonlanan bir NURBS olmalıdır. (Şekil 4.2, Şekil 4.3)



Şekil 4.2 İlk eskizin analitik soyutlaması



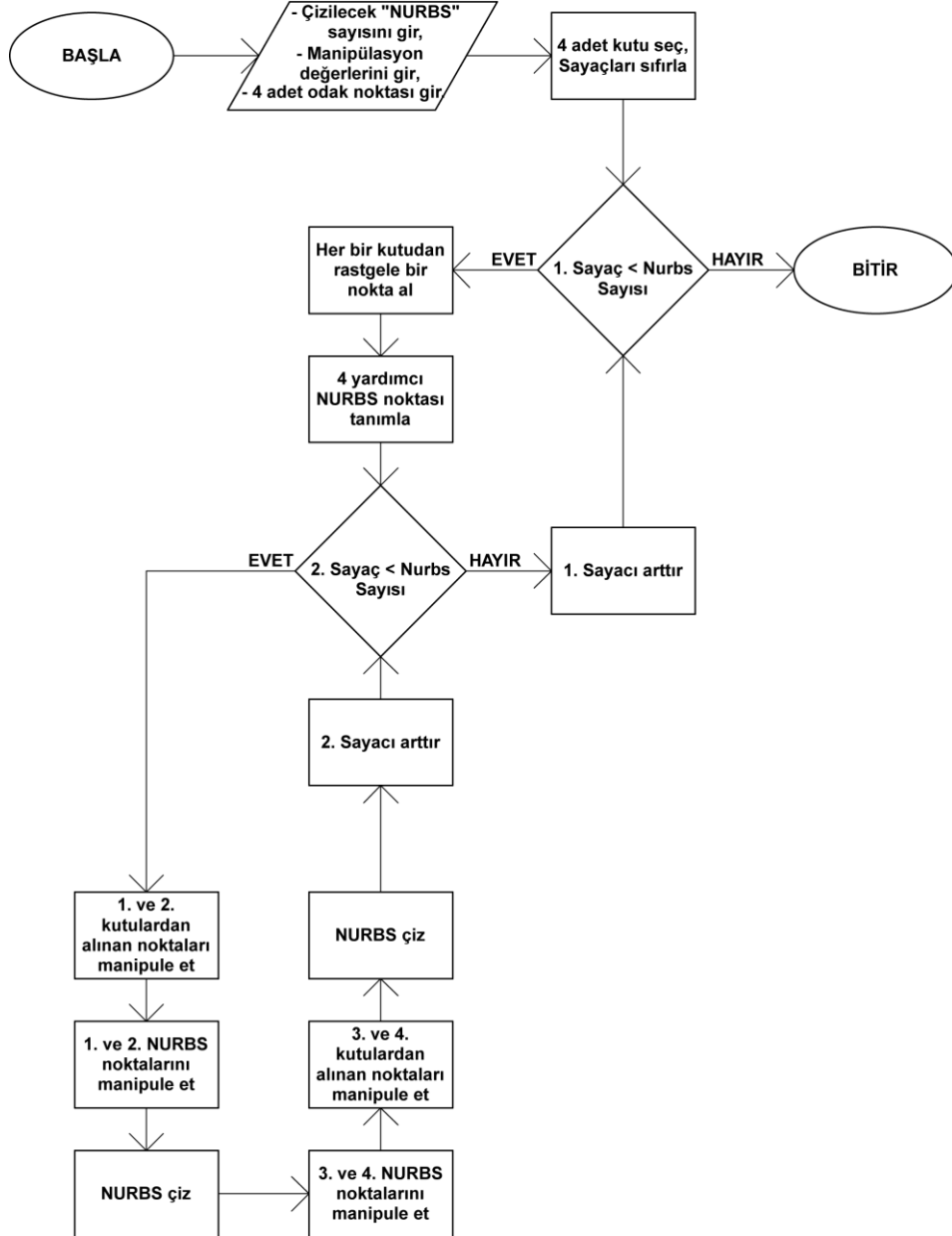
Şekil 4.3 Fikrin analitik ve algoritmik soyutlanması

Son olarak belirlediğimiz iki odak noktası ve iki üst yüzey noktası NURBS çizme amaçlı kullanılmaktadır. Analitik değerleri her döngüde değiştirilmediği takdirde ancak tek bir NURBS'ü defalarca çizilecektir. Bu sebeple öncelikli olarak kütlelerin üst yüzeyinden alınan M ve N noktalarının rastgele yüzeyden herhangi bir nokta seçilmesi yoluna gidilmiştir.

Eskizi kabaca analitik ortamda tekrar düşünmenin yararı problemin parametrik hale getirilmesi olmaktadır. Parametreler dahilinde fikrin, algoritmik bir fikir olarak dışavurumu için ilk adım olmaktadır. Algoritmik fikir olarak bu kütlelerin üzerini örtmek için, bu fikir algoritmik olarak düşünülmelidir. Adımlanmalıdır.

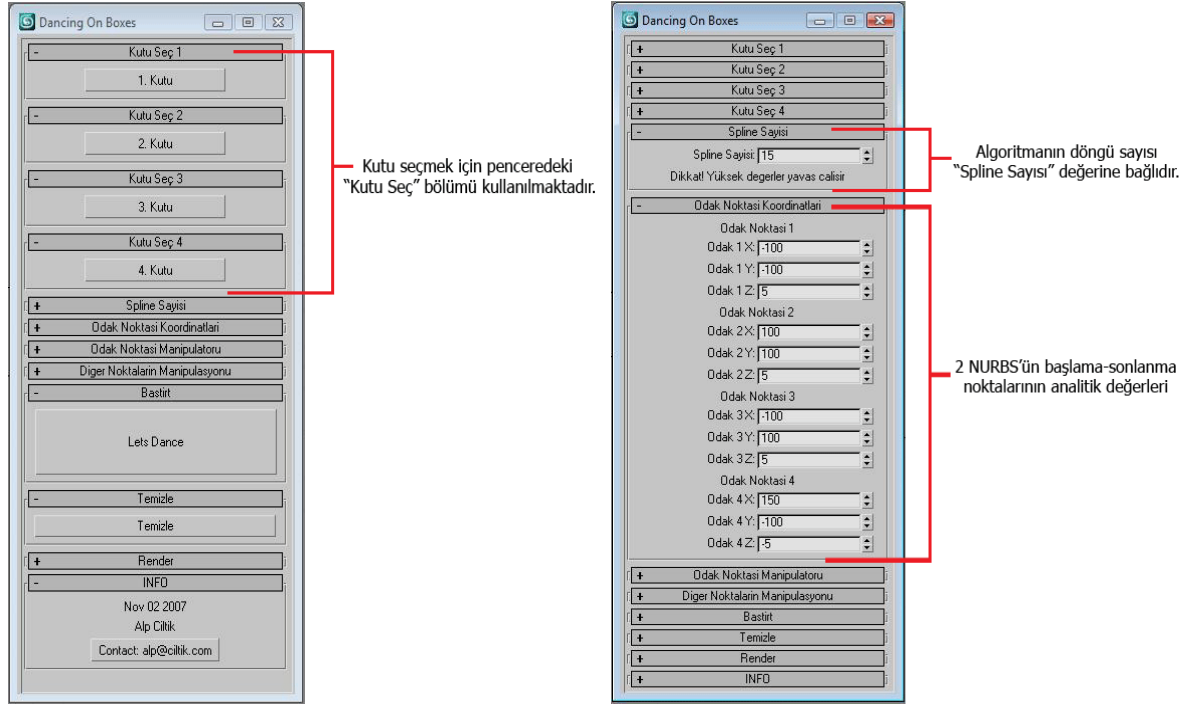
- A(XYZ) --A noktasından NURBS çizmeye başla.
- M(XYZ) --İlk kütlelenin üzerinden bir nokta seç.
- N(XYZ) -- İkinci kütlelenin üzerinden bir nokta seç.
- B(XYZ) -- B odak noktasında NURBS'ü bitir ve tekrar başla.

Bu algoritma şematize edilirse, aşağıdaki akış diyagramı çizilebilir. (Şekil 4.4)



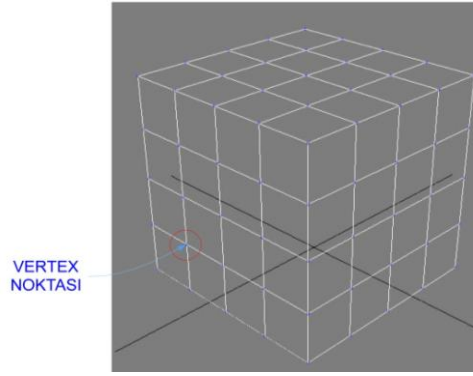
Şekil 4.4 Yazılan algoritmanın akış diyagramı

Bu algoritma, parametrik değerleri değiştirilmez ise hep aynı sonucu verecektir. Kurulmuş algoritmanın parametreleri, dört noktanın analitik değerleridir. Algoritmanın sonsuza kadar devam etmemesi için de ayrıca bir döngü sayısı verilmelidir. Düşünülen döngü sayısı ve odak noktası koordinatları script için oluşturulmuş arayüzde girilmelidir.(Şekil 4.5)



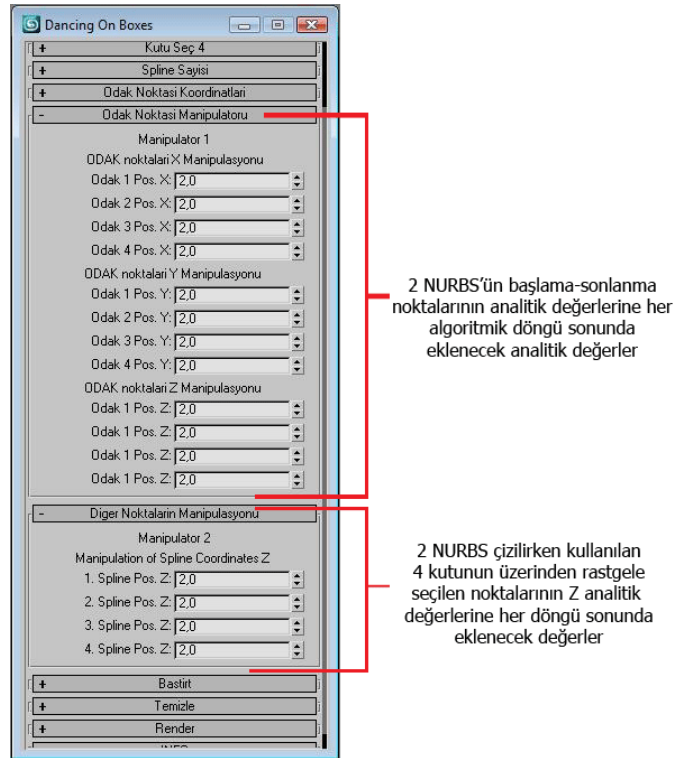
Şekil 4.5 Script arayüzü

Bina kütleleri için MaxScript'in tasarımcıya verdiği bir kolaylıktan yararlanarak analitik fonksiyonu yazmak yerine sadece "BOX" denebilmektedir. Bu sayede referans olarak "BOX" nesnesini alınabilmektedir. Bu nesnenin üst yüzeyinde bulunan noktaların analitik değerleri üst örtünün çizilmesi sırasında kullanılacaktır. Scripti çalıştırdıktan sonra istediğimiz kutuyu fare oku yardımıyla ekrandan seçtiğimizde kutu yüzeylerine otomatik olarak belirli sayıda vertex atılmaktadır. Başka deyişle; kutunun yüzeylerine belirli sayıda gridal aks sistemi yerleşmektedir. Kutunun üst yüzeyinin gridal aksların kesişim noktaları, M üst noktasını analitik olarak belirtmemizi sağlamaktadır. (Şekil 4.6)

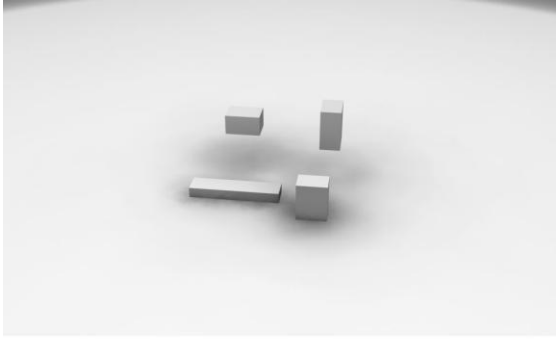


Şekil 4.6 Grid kesişimlerindeki vertex noktaları

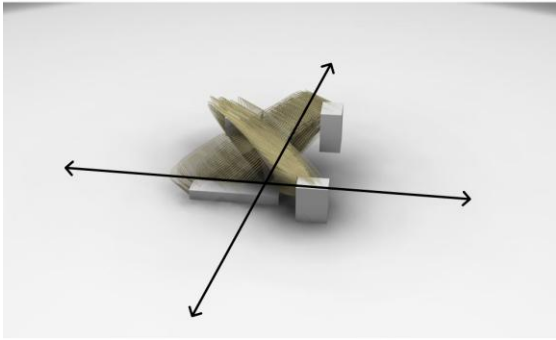
Tasarımı geliştirmek amacıyla, A ve B noktalarına ek olarak C ve D odak noktaları da eklenerek “A, $M(rastgele)$, $N(rastgele)$, B” birinci NURBS’ün yanı sıra “C, $M(rastgele)$, $N(rastgele)$, D” ikinci bir NURBS de aynı anda çizdirilmektedir. Her döngüde bu dört noktayı ayrıca değiştirebilmek amacıyla önce odak noktalarının XYZ değerlerine eklenmek üzere birer manipülasyon değeri (Şekil 4.7) ve $M(rastgele)$, $N(rastgele)$ Z değerine eklenmek üzere bir birer manipülasyon değeri belirlenmektedir. Çizilen her bir NURBS birbirlerine bu manipülasyon değerleri üzerinden bağlıdır.(Şekil 4.8, 4.9)



Şekil 4.7 Manipülasyon değerleri

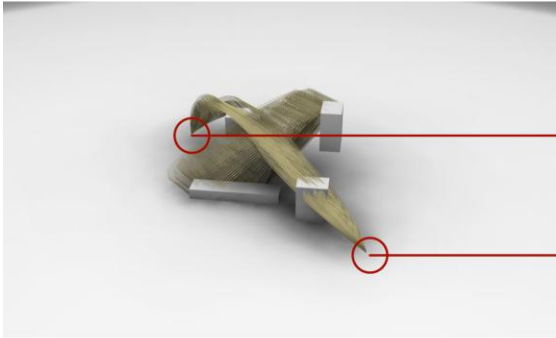


Olası Kütle organizasyonu

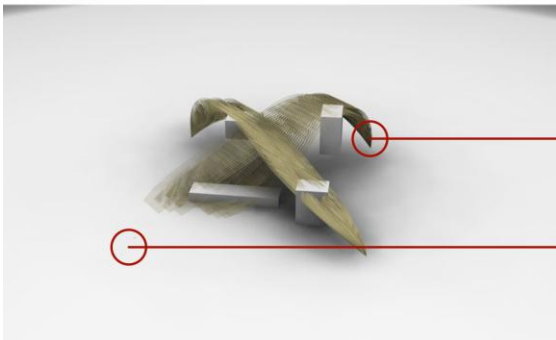


Tüm odak noktalarının koordinatları XYZ (0,0,0) noktasındadır

Algoritma M ve N noktalarını kütlelerin üst yüzeylerinden rastgele seçmektedir.

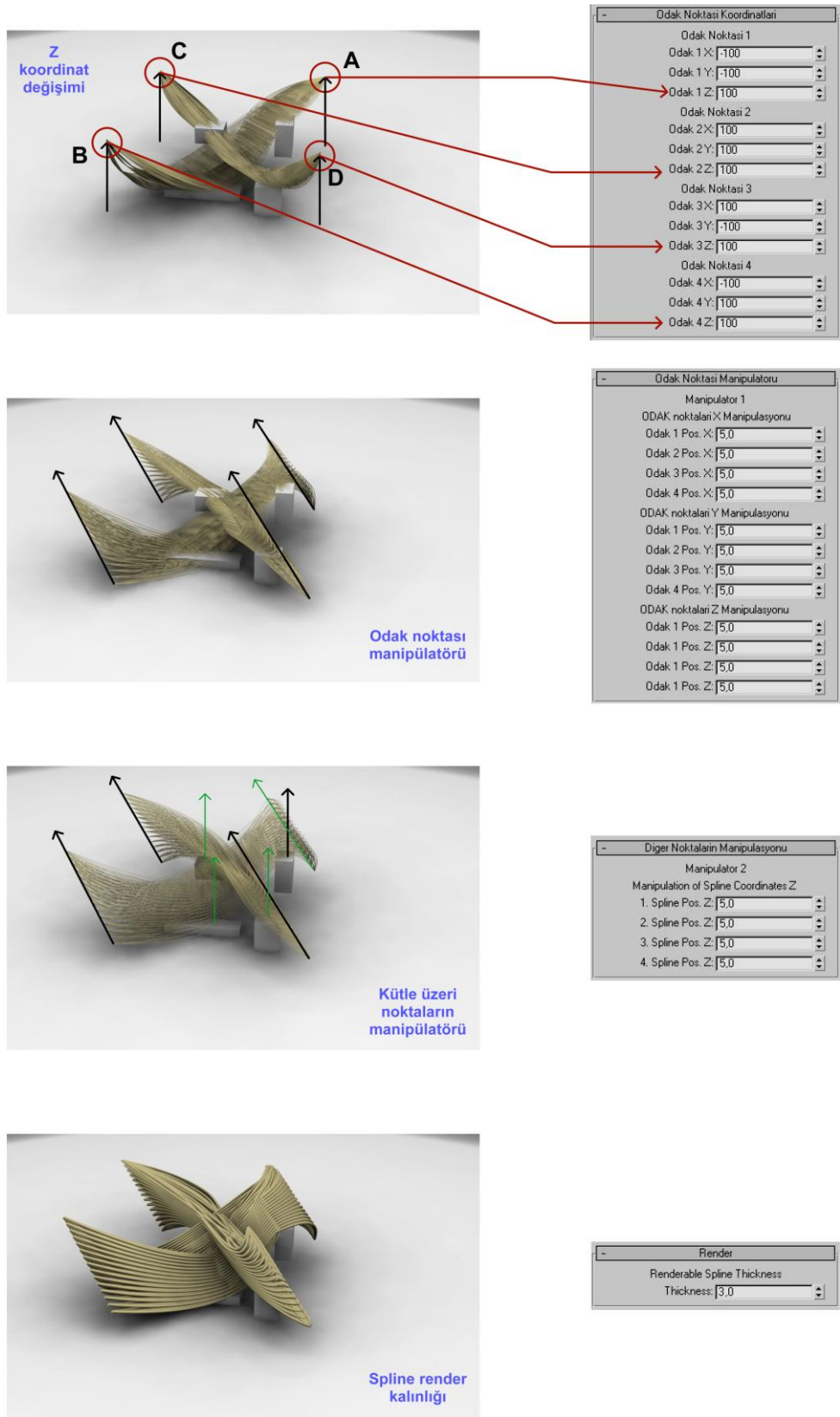


Odak Noktası Koordinatları	
Odak Noktası 1	
Odak 1 X:	-100
Odak 1 Y:	-100
Odak 1 Z:	0
Odak Noktası 2	
Odak 2 X:	100
Odak 2 Y:	100
Odak 2 Z:	0



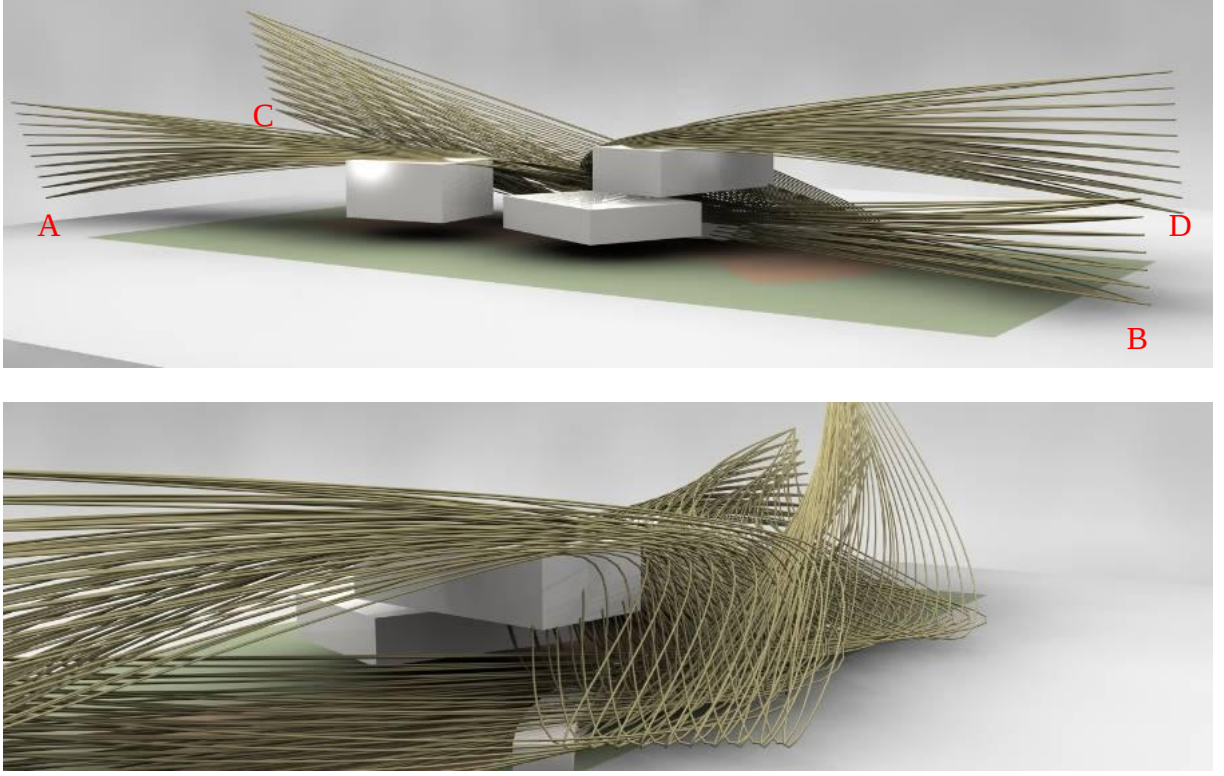
Odak Noktası Koordinatları	
Odak Noktası 3	
Odak 3 X:	100
Odak 3 Y:	-100
Odak 3 Z:	0
Odak Noktası 4	
Odak 4 X:	-100
Odak 4 Y:	100
Odak 4 Z:	0

Şekil 4.8 Script kullanımı sırasında odak noktaları belirlenmesi



Şekil 4.9 Script kullanımı sırasında diğer manipulatorlerin belirlenmesi

Deney öncesi süreçte denenen script ile çeşitli ürünler elde edilmiştir (Şekil 4.10). Üretilen ürünler çeşitli odak noktaları ve manipülasyon değerleri denenerek elde edilmiştir. Dört odak noktası ve kütleler üst yüzeylerinden rastgele alınan ara noktalar bir eğri ile birleştirilerek sonuç ürünler elde edilmiştir. Eğri sayısı kalınlığı script arayüzünde değiştirilebilmektedir.



Şekil 4.10 Deney öncesi ortaya çıkabilecek ürün araştırmasından örnekler

4.2 Deney Süreci

Program yazıldıktan sonra sekiz kişilik mimardan oluşan bir grupla Yıldız Teknik Üniversitesi, Bilgisayar Ortamında Mimarlık Atölyesi'nde yarım günlük bir atölye çalışması yapılmıştır. Bu atölye çalışmasındaki amaç; algoritmik düşünceyle hazırlanmış bir yazılımın kullanımı üzerine katılımcı mimarlardan yorum almaktır. Katılımcılardan yorumlar ve eleştiriler alınmış ve tasarım sürecinin nasıl gelişeceği, bu yazılımın tasarım sürecine nasıl katılacağı üzerine kısa bir tartışma ortamı oluşturulmuştur. Atölye çalışması Bilgisayar Ortamında Mimarlık programının laboratuvarında uygulanmıştır. Katılımcılar dört bilgisayarı ikiye ikiye dönüşümlü olarak kullanmışlardır (Şekil 4.11, 4.12).

Katılımcı profili 3dsMax arayüzüne aşina yüksek lisansına devam eden veya bitirmiş mimarlar arasından seçilmiştir. Script kullanımı için 3dsMax bilme zorunluluğu yoktur.

Katılımcıların içinde MAXscript dili kullanabilen katılımcının yanı sıra script yazımı hakkında fikri olan toplam dört katılımcı vardır. Diğer dört katılımcı script yazımı hakkında fikir sahibi değildir. Amaçlanan yöntem içinde değerlendirmeden çok, bu yöneme farklı açılardan bakmaktır. Bu sebeple yazılmış bir script üzerinden yapılacak bir atölye çalışması yoluna gidilmiştir. Mimarlar lisans eğitimlerini Mimar Sinan, Yıldız Teknik Üniversitesi yanı sıra İzmir Yüksek Teknoloji Enstitüsünde tamamlamış bir gruptur.



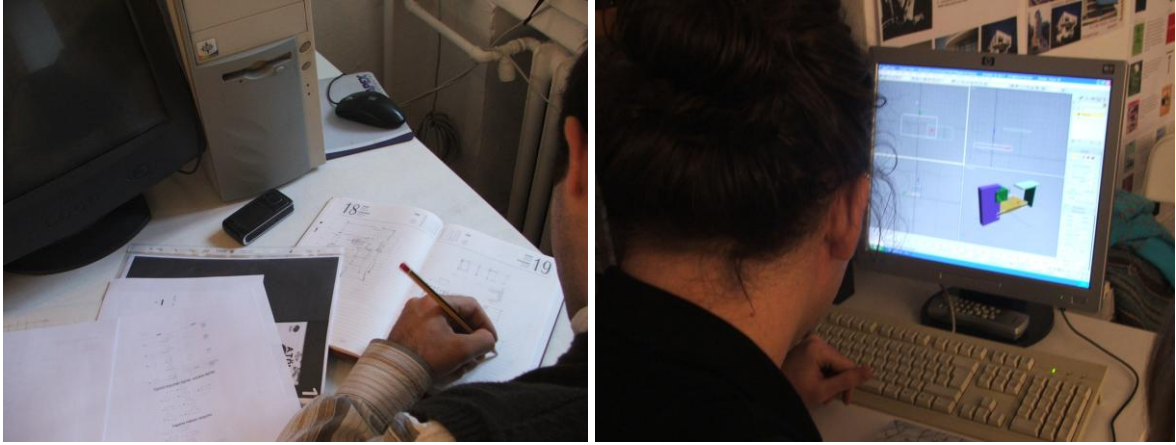
Şekil 4.11 Atölye çalışmasından görünüm



Şekil 4.12 Atölye çalışmasından görünüm

Çalışma üç temel kısımdan oluşmuştur. Birinci kısımda bağımsız eskizler, ikinci kısımda program kullanımı ve üçüncü kısımda ise program verileri kullanılarak eskizler istenmiştir.

Bu adımlardan sonra bir tartışma yapılmış ve kullanıcılardan yorumları alınmıştır (Şekil 4.13).



Şekil 4.13 Atölye çalışmasından eskiz aşaması ve üç boyutlu ürün üretimi aşaması

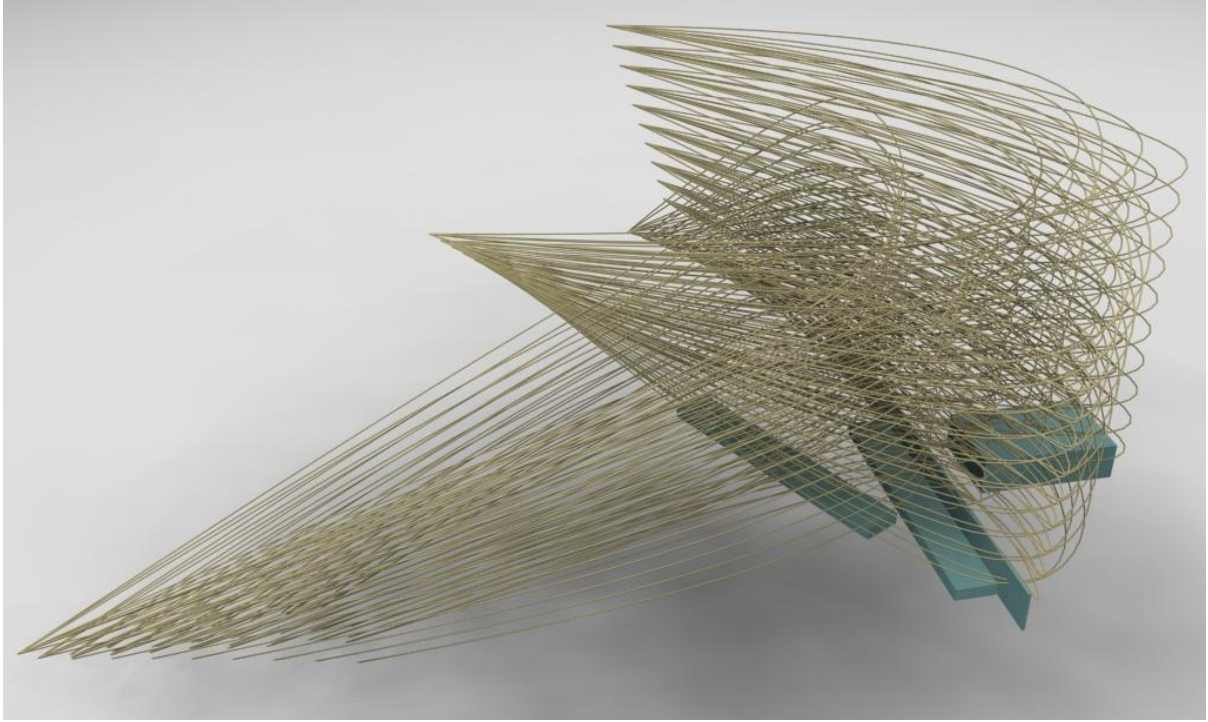
“Script” 3dsMax’te “MAXScript” menüsüne girilerek “run script” komutuyla yazılmış olan algoritmanın çalıştırılmasıyla elde edilmektedir. Programın arayüzünde kutuları seçmek için butonlar, eğri (NURBS) sayıları, odak noktalarının koordinatları ve manipülatörleri, diğer noktaların manipülatörleri yanı sıra render aşamasında Nurbs’lerin kalınlığı ve aynı kütle organizasyonu tekrar kullanabilmek için oluşturulan Nurbs’leri silme butonu bulunmaktadır.

Çalıştırılan programda öncelikle ekranda dört tane kutu seçilmesi gerekmektedir. Bu kutuların seçilmesiyle; seçilen kutuların her birinin üzerinde, her yönde eşit aralıklarla dörder adet noktası (segment) yazılım tarafından işaretlenmektedir (Şekil 4.4). Bu noktalar türetme yapıldığında eğrilerin üzerlerinden geçeceği noktalardır.

Daha sonra “Let’s Dance” butonuna basarak gerekli türetmenin yapılması sağlanmaktadır (Şekil 4.14). Böylelikle ekrana türetilen ürün basılmaktadır. Ardından manipülatörler kullanılarak, parametrelerle oynanarak aynı ürünün çeşitli fenotipleri elde edilebilmektedir

İlk aşamada katılımcılardan farklı boyut ve işlevlere sahip olabilecek binalardan oluşan bir yapı kompleksi için kütle organizasyonu istenmiştir. Bu aşamada katılımcının minimum girdi ile serbest eskiz yapması amaçlanmıştır. Ardından eskizle yapılan bu çalışma sonrasında katılımcılardan 3dsMax’te “script” (Ek 1) kullanılarak mevcut arazide konumlanacak kütlelerin üzerini örten bir kabuk tasarımı üretmeleri istenmiştir. Bunun için oluşturdukları kütle organizasyonlarını 3dsMax ortamına aktarmışlardır. Üçüncü aşamada ise üretilen

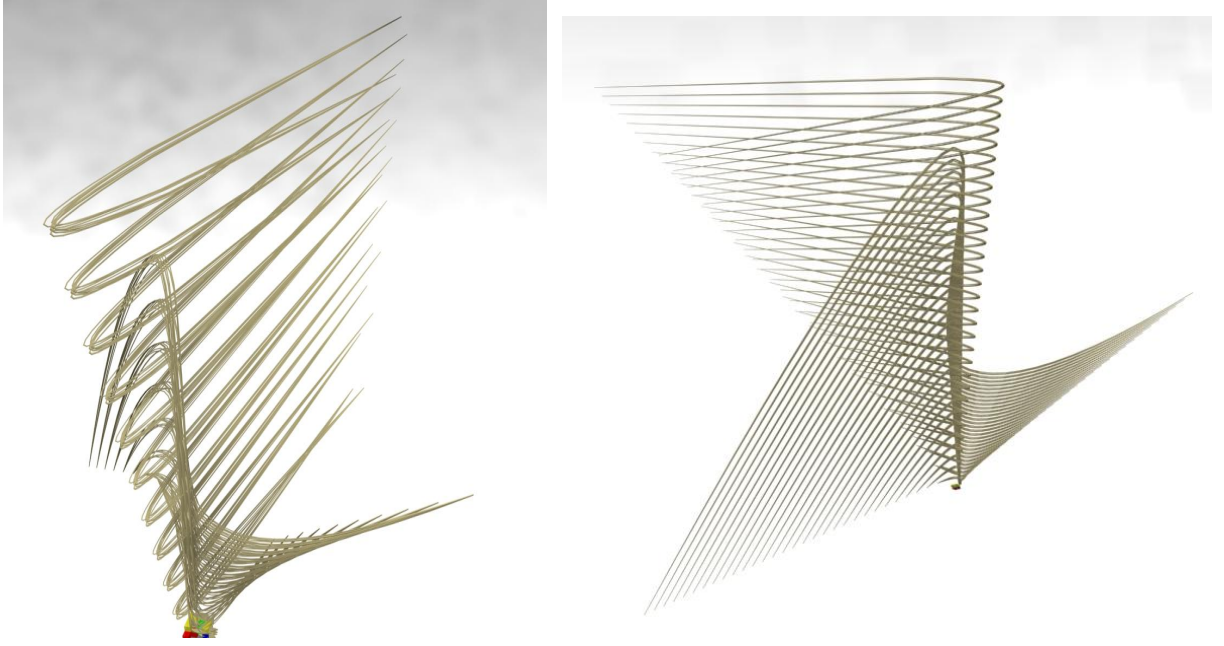
ürünlerin üzerinden yeni bir eskiz çalışması daha istenmiştir. “Script”in parametreleri ile oynanarak çeşitli ürünler elde edilmiştir (Ek 2). Son olarak ise programdan elde edilen ürünler göz önüne alınarak ikinci eskizi yapmaları ve genel olarak bu süreçleri yorumlamaları istenmiştir (Ek 3)



Şekil 4.14 Atölye çalışmasından üç boyutlu ürün üretimi

Belirlenen aşamalar için kullanıcılara her aşama için 20 dakika verilmiştir. Katılımcılar özellikle scriptin denendiği aşamada verilen süreyi fazlasıyla aşmıştır. Bunun ilk sebebi scriptin algoritmik alt yapısının sunuma rağmen derhal anlaşılamamasıdır. Sonrasında üretilen ürünlerin beklediklerinden farklı olması durumudur. Son olarak ise scripte hakimiyet kurulması ve istenene yakın ürünler üretilmesi süreci gelmiştir. Bu sebeple ikinci aşama verilen sürenin üzerinde bitirilebilmiştir. Kimi kullanıcılar ilk eskiz aşamasını geçmek istemiş derhal Max ortamında kütle organizasyonuna yönelmiştir. Bu kullanıcılardan yine de eskiz çalışması beklenmiştir. Kimi kullanıcılardan ise ikinci eskiz aşaması sırasında verim alınamamıştır. Bazı kullanıcılar ikinci eskizlerinde kütle organizasyonu ile sentez yerine tamamıyla farklı soyutlamalarla eskizler üretmiştir. Genellikle kullanıcılar script kullanımı üzerine ürettiği ürünlere yönelim göstermiştir. Bunların yanı sıra ikinci eskizinde birinci ve ikinci aşamalarda ürettiği ürünlerle senteze yönelen kullanıcılarda vardır (Şekil 4.16).

Ayrıca parametreler script yazılırken belirli aralıkta tutulmuştur. Buna rağmen atölye sırasında kullanılan dört bilgisayarın birinde hali hazırda kurulmuş programın birim ünite ayarı farklı olmasından dolayı kütleden bağımsız ürünlerden elde edilmiştir. Bu bilgisayar kullanıcılarına, ayarların değiştirilmesi önerildiğinde bunun gerekli olmadığı belirtmişlerdir. Bu şekilde çıkan ürünlerin, sadece kütle organizasyonuna üretilen bir kabuk gibi algılanmadığını belirtmiştir. Farklı bakıldığında bu scriptin sadece kabuk üretmek için değil farklı amaçlara farklı ürünlere de açık olduğunu belirtmişlerdir (Şekil 4.15).



Şekil 4.15 Atölye çalışmasından üç boyutlu ürün üretimi

Script içeriğinde yer alması düşünülen fakat gerçekleştirilemeyen çeşitli unsurlarda bulunmaktadır. İlk olarak kütle organizasyonu üzerinde oluşturulan Nurbs'lerin parametrelerini kaydetmek için “kaydet” ve karşılaştırma yapabilmek için “klon” butonlarının da script'e eklenmesi düşünülmüş fakat gerçekleştirilememiştir. Bunun yerine klonlamak istendiğinde Max dosyasının farklı isimle kaydedilmesi yoluna gidilmiştir. Diğer bir eksik ise bir önceki ürüne tek tuş yardımıyla geri dönüşün olmamasıdır. Bu eksik de ismi farklı kaydedilen dosyanın tekrar açılması yoluyla kapatılmıştır.

4.3 Deney Sonucu

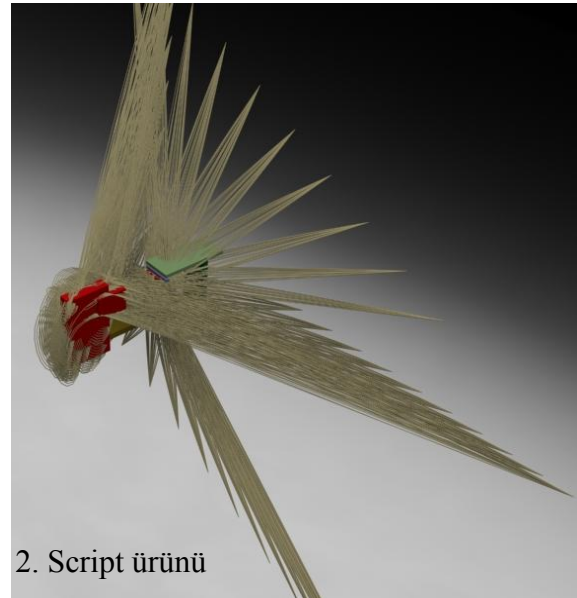
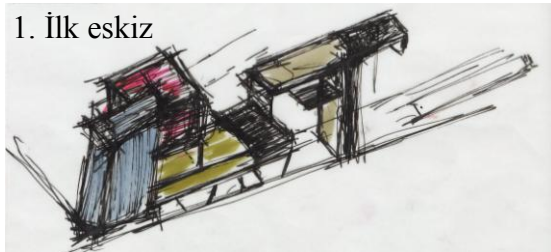
İlk eskiz çalışması, tasarımcının problemi ilk çözmeyi düşündüğünde, çözüm üretmek için bugüne kadar kullandığı akıl yürütme eyleminin dışavurumudur. Günümüzde mimarlar

yetiştirilirken, fikrin dışavurumunun en kolay yöntemi olan kağıt, kalem ve eskiz birlikteliği kullanılmaktadır. Yapılan eskiz, tasarımcının tasarısını ilk etapta en kolay dışavurum biçimidir. İlk eskiz aşaması algoritmik düşünceyle beraber tartışmayı amaçlayan deneysel alan çalışması sonucunda;

—İlk eskize başlarken kullanıcıların tamamı öncelikli olarak tasarım probleminde verilen girdileri kullanmakta olduklarını belirtmişlerdir. Tasarım yaptıkları uzayın boyutlarını algılama ve bu uzayı gerektiği kadar daraltma çabası içinde olduklarını belirtmişlerdir.

—Eskiz yapma sürecinde kendilerini etkileyen ikincil kavramların ortaya çıkışını sezgisel olarak gören bir grup yanı sıra, kendi mimari beğenileri ve mekan oluşturma kaygıları üzerinde durduklarını söyleyen diğer bir grup da vardır.

—Yazılımı deneme sürecinde ise; kullanıcılar programlama diline hakim olmadıklarından dolayı ortaya çıkan ürünleri öncelikle tahmin edilemez görmektedir. Kullanıcıların, özellikle yazılımı incelemeye devam eden kısmı, zaman harcandığında yazılımın üretebileceği ürünleri daha kolay algıladıklarını ve bunun zamana ve yazılıma alışmak ile ilgili olduğunu düşünmektedirler.



Şekil 4.16 Bir kullanıcının deney sürecinin adımları (EK 2)

—Genellikle yazılımın kullanımının kolay olduğu görüşünde olan grup zorlanmanın ara yüzden ziyade matematiksel ve analitik düşünmeye alışkın olmamaktan kaynaklandığını belirtmektedirler.

—Tasarım sürecinde yazılım kullanımı ile ilgili olarak ise; bir kısım kullanıcı programlama dillerine olan uzaklıkları sebebiyle yazılım üreterek tasarım yapmanın el eskizi ile yapmaktan çok daha sıkıntılı bir süreç olacağını düşünmektedirler. Parametrelerin ekrana bu hızda gelmesinden etkilenen kimi kullanıcılar, programlama yazımının mimarın dilini daha kolay aktarabileceği bir durum söz konusu olduğunda veya yazılım yazmayı kolaylıkla yaptığında kesinlikle programlama dillerinin kullanılabileceği fikrindedirler.

—Yazılım kullanımının ardından yapılan ikinci eskiz aşamasında ise; bir grup kullanıcı daha önce düşünmediğini veya beklemediğini söyledikleri yazılım ürünleri yardımıyla tasarımı tekrar düşündürdüğünden bahsetmişlerdir. Bu tekrar düşünme sürecince yazılımı yazan kişinin fikirleriyle ortaklaşa kurguya gittiğini belirten kullanıcıların yanı sıra algoritmik ürünün ritmik yapısından etkilendiğini belirten kullanıcılar da vardır. Kullanıcılar yazılım ürünlerinin tasarımlarını formal açıdan etkilediklerini belirtmişlerdir. Düşündükleri rasyonel çizgilerini yumuşatmaya kırmaya çalışmaktadırlar. (Şekil 4.16)

Alan çalışmasının genelinden elde edilebilecek en önemli sonuç, mimarların algoritmik düşünce biçimine çok ilgisiz olmadıkları ve günümüze kadar mühendislik çalışmaları için üretilen bilgisayar dil sistemlerinin yanı sıra mimarlara özel dil kodlama çalışmalarının yapılması gerekliliğidir. Bu sayede mimarların kodlamaya çok daha hızlı entegre olarak daha rafine ürünler üretmesi beklenmektedir.

5. DEĞERLENDİRME VE SONUÇLAR

Mimarlık disiplini, insanın yaşam kalitesini artırma çabasıyla tasarım ürünlerini daima geliştirmektedir. Tasarım ürünlerinin gelişmesi için tasarım teknikleri ve fikir alt yapısı daima araştırılmıştır. Gelecekte de araştırılacak bir konu olacaktır. Mimarlık insan yaşam kalitesini arttırmak amacı güden bir disiplindir. Bu sebeple mimari tasarım ürünü felsefe, sosyal, ekonomi ve kültür dokularının değişiminden fazlasıyla etkilenmektedir.

Devrim ve eşik dönemleri esnasında, felsefe ve pozitif bilimlerin iletişimi, insanın yaşam biçimi ve kalitesini etkilemektedir. Yaşam ile birebir bağlı bir disiplin olan mimarlık disiplini bu değişimlerden fazlasıyla etkilenmektedir. Mimarlık disiplini, teknik ve fikir gelişmelerinin yaşandığı eşik dönemlerde değişmektedir. Bu değişimleri Eski Yunan uygarlıkları, Roma İmparatorluğu, Rönesans, Endüstri devrimi benzeri eşik dönemler sonrasında gözlemlemek mümkündür.

Günümüzde mimarlık, 20. yüzyılın sonunda ortaya çıkan sayısallaşma, küreselleşme, sürdürülebilirlik kavramlarından etkilenmektedir. Sayısallaşma, mimari ürünün tasarımdan üretimine kadar sürecin sayısal ortama geçmesine sebep olmaktadır. Küreselleşme, ürünü üretme sürecinin çeşitli safhalarının dünyanın farklı noktalarında gerçekleştirilmesine izin vermektedir. Mimarlık disiplini doğaya uyumu üst seviyede olacak ürünler üretebilmek için doğanın verdiği parametrelere daha çok önem vermektedir. Evrim ve doğanın parametrelerine göre kendini organize edebilen mimari tasarım ürünlerini üretebilmek için mimari ürüne genetik yapay organizmalar olarak bakılmaktadır. Bu kavramlar iç içe girmiş ve devamlı etkileşerek gelişmektedir. Bilgi akış hızının artmasıyla bu gelişme, mimarlık disiplini adına yeni bir dönem olarak görülmektedir.

Mimarlık disiplini açısından bilgisayar fikrinin geçmişi, özellikle son çeyrek yüzyıla yayılmaktadır. 1960'larda bilgisayar ve mimarlık iletişimi üzerine araştırmalar olsa da 1980'lerin başından itibaren, bilgisayar, mimarlık sektöründe yer almaktadır. Bu yıllarda BDMÇ aracı olarak kullanılmaktaydı. BDMÇ sürecinde kalem ve kağıdın, fare ve ekran ile taklit edildiği geleneksel BDT kavramları, mimarların bilgisayarla iletişiminin son sınırı olmaktaydı. Bu süreçte mimari projeler “analog temsili ortam”dan “dijital temsili ortam”a geçmiştir. Üretim sürecindeki ve ürün temsilindeki dijitalleşme, tüm mimarlık alanlarına baskıda bulunmaktadır. Mimari tasarımında bu süreçte dijitalleşmesi gündemdedir.

Mimari sözlük olarak düşünülen kapılar, pencereler, duvarlar ile soyutlanmış sayısal benzerleri arasında doğrudan bir haritalama söz konusudur. Mimarlar BDMT ve BDMÇ paket programlarında mimarların eline verilen sözcük sayısı, paket programı üreten firma tarafından belirlenmektedir. Güncel parametrik tasarım araçlarının farkı ise, bu “sözcüklerin” daha soyut ve aynı zamanda sistemin daha genişletilebilir olmasıdır. Böylece bileşenlerin sözlüğünü üreten tasarımcının kendisi olur. Bu sayede programın verdiği kapalı uçlu çalışma diliyle değil, tasarımcının kendine ait diliyle anlaşılan ürünler elde edilebilir. Tasarımcı kendine ait sözlük tanımlama fırsatına sahiptir.

Tasarım kavramının *kompleks* kavramıyla birlikte düşünüldüğünde, tasarım sürecinin kişiye özgü çözümlemesi için, *algoritmalar* bilim adamları tarafından öne sürülmektedir. Günümüzde hemen hemen tüm algoritmik bilgisayar dilleri, farklı mühendislik dalları için geliştirilmekteydi. Algoritmik düşüncenin dillendirilmesi kişiye özgün bir yoldur. Günümüzde BDT programı üreten firmalar bu ilerlemenin farkındadır. Mimarlar için algoritmik dil oluşturma çabası henüz akademik bir çalışma alanıdır. Akademik ortamda yeni programlar oluşturmaya çalışmaktadırlar. Günümüzde bu amaca yönelik paket programlar henüz gündeme gelmektedir. Bu teknolojileri ne kadar yeni olduğuna örnek olarak Bentley firmasının desteğiyle 2000’li yılların başından beri akademik ortamda geliştirilen parametrik tasarım programı “GenerativeComponents” henüz Kasım 2007 içerisinde internetten deneme sürümü olarak yayınlanmaya başlamıştır. Zamanla algoritmik düşüncüyü içine sindirmiş ve kendi içinde esnek, daha çok paket programın yayımlanması beklenmektedir.

Tez içerisinde yürütülen alan çalışması öncesinde yazılan yazılım (script), tasarım bilgisayar entegrasyonuna küçük ölçekte bir örnek olmaktadır. Alan çalışmasının sonuçları değerlendirildiğinde, kimi katılımcıların script kullanımına yabancı kaldığı gözlemlenmiştir. Bu gözlemin yanı sıra parametrik ve algoritmik düşünce biçiminin ortaya koyduğu potansiyelin varlığı yadsınamamaktadır. Alan çalışmasına katılan mimarların fikirleri değerlendirildiğinde, mimarlar, alışkanlıkların ve verilen eğitim biçiminin dışına taşmakta temkinli davranmaktadır. Birçok katılımcı algoritmik ve analitik düşüncenin kendi yöntemlerine uzak olduğu belirtmektedirler. Mimarlar henüz bilgisayarları, sadece mimari teknik dili daha kolay ortaya dökmeye yarayan bir araç olarak görmektedir. Aslında bilgisayar, hesaplama gücünün yanı sıra algoritmik düşünce mantığının merkezinde yer almaktadır.

Katılımcılar, fikirlerini daha kolay yazılıma dökebilecekleri bir dillenme olması halinde script yazımını kullanabileceklerini söylemektedirler. Alan çalışması için yazılan scriptin yetersizliğini savunan katılımcılar ise fikirlerini istedikleri gibi dışa vuramadıklarını söylemektedirler. Bu katılımcılar, scriptin, yazarının isteklerine dair olduğunu belirtmişlerdir. Script yazmak tasarım sürecinin değerlendirilmesidir. Her tasarımcı kendi sürecini kendisi değerlendirmelidir. Bu noktada, her tasarımcının, tasarlayacağı her ürün için, farklı birer script üretmesi gerekmektedir. Tasarımcılar fikirlerinin algoritmasını kendileri yazması daha verimli olacaktır. Bu sebeple, günümüze kadar mühendislik dalları için üretilen bilgisayar dil sistemlerini kullanmak yerine tasarımcıların kolaylıkla kullanabileceği, tasarımcılara özel programlama dili üretilmelidir. Mimarlık disiplinin dilini, kökten değiştirmek mümkün değildir fakat gelişmeler tasarımcı profili ve dilinin değişeceği ön görülmektedir.

Atölye çalışmasında da deneylendiği gibi, tek bir “script”le *farklı* ürünler elde edilmiştir. Bu ürünlerin hepsi temelde aynı algoritmadan türetilerek aynı genetik *uygunluğa* sahiptir. Bununla birlikte birbirinden *farklı* ürünler ortaya konulmuştur. Sonuçta bir algoritmadan tek bir ürün değil, ortak temel özelliklere sahip bir ürün grubu ortaya konabilmektedir. Bu açıdan bakıldığında tasarımcının algoritmayı tasarlaması; bu ürün grubunu ortaya koymak için gerekli kuralları ortaya koyması anlamına gelmektedir.

Çeşitli parametreleri kullanarak ve bunları manipüle ederek ortaya konan ürünler, bilgisayarın sadece projenin son döneminde proje sunmak için kullanılacak bir aracın ötesine geçtiğinin kanıtıdır. Bahsedilen değişimler, mimarların tekrar matematiğe sarılması gerekliliğini göstermektedir. Mimarlar tasarım arkadaşı olarak bilgisayarları yanlarına almak zorunda kalacakları açıktır. Bu da disiplinler arası ilişkinin daha üst noktalara taşınacağı anlamına gelmektedir. Zamanla bu da yetersiz kalarak “*disiplin ötelemesi*” yaşanması olasıdır.

Mimari tasarım süreciyle entegre olan bilgisayar ve sayısal teknolojiler gün geçtikçe tasarım ve üretim süreci, analog bir yapıdan dijital bir kimliğe bürünmektedir. Sonuçta mimari ürün üretme sürecinin *en başından en sonuna* kadar “*dijital bir eylem*” olması öngörülmektedir. Mimarlığın dijitalleşmesi sürecinde mimarlar analog teknikleri örneğin kağıdı kalem malzemeyi ne kadar iyi biliyor ise gelecekte de dijital ortamı analog ortam kadar iyi bilmesi gerekmektedir. Bu sayede daha rafine ürünler üretilebilecektir. Günümüzde dijital ortam ve kavramları, henüz algılanmaya çalışılmaktadır. Bu mimarlar için yeni bir alandır.

KAYNAKLAR

- Akipek, F., “*Bilgisayar Teknolojilerinin Mimarlıkta Tasarımı Geliştirme Amaçlı Kullanımları*” Yıldız Teknik Üniversitesi, Bilgisayar Ortamında Mimarlık, Doktora Tezi, İstanbul 2004
- Bayazıd, N., “*Tasarlama Kurguları ve Metodları*”, Birsen Yayınevi, 2004
- Burke, A., (2006), “*After BitTorrent: Darknets to Native Data*”, Architectural Design, 76 Issue 5 September/October, 88-95.
- Burrow, A., and Burry, J., (2006), “*Working with Wiki, by Design*”, Architectural Design, 76 Issue 5 September/October, 97-99.
- Chu, K., (2006), “*Metaphysics of Genetic Architecture and Computation*”, Architectural Design, 76 Issue 4 July/August, 38-45.
- Deleuze, G. “*Difference & Repetition*” The Athlone Press:London 1994, p.182
- Deleuze G. and Guattari F. “*A Thousand Plateaus*”, University of Minnesota Press, Minneapolis, 1987 p.48
- Ednie-Brown, P., (2006), “*All-Over, Over-All: biothing and Emergent Composition*”, Architectural Design, 76 Issue 4 July/August, 72-81.
- Ednie-Brown, P., and Andrasek, A., (2006), “*CONTINUUM: A Self-Engineering Creature-Culture*”, Architectural Design, 76 Issue 5 September/October, 19-25.
- Erdman, D., Gow, M., Karlsson, U., and Perry, C., (2006), “*Parallel Processing: Design/Practice*”, Architectural Design, 76 Issue 5 September/October, 81-87.
- Galloway, R. A., and Thacker, E., (2006), “*Language, Life, Code*”, Architectural Design, 76 Issue 5 September/October, 26-29.
- Gardner M. (1970) “*Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"*” Scientific American 223 (October 1970): 120-123
- Goldenberg, J., Horowitz, R., Levav, A., and Mazursky, D.; “*Finding Your Innovation Sweet Spot*”, Harvard Business Review, 2003
- Goulthorpe, M., and dECOi, “*Scott Points: Exploring Principles of Digital Creativity*”, “*Architecture in Digital Age: Design and Manufacture*” Spon Press, London, 2003
- Harel, D. (1992). “*Algorithmics: the spirit of computing (2nd ed.)*.” Reading, MA: Addison Wesley.
- Hensel, M., (2006) “*(Synthetic) Life Architectures: Ramifications and Potentials of a Literal Biological Paradigm for Architectural Design*”, Architectural Design, 76 Issue 2 March/April, 18-25.
- Hensel, M. and Menges, A., (2006), “*Differentiation and Performance: Multi-Performance Architectures and Modulated Environments*”, Architectural Design, 76 Issue 2 March/April, 60-69.
- Hensel, M. and Menges, A., (2006), “*Material and Digital Design Synthesis*”, Architectural Design, 76 Issue 2 March/April, 88-95.
- Hensel, M., (2006a), “*Computing Self-Organisation: Environmentally Sensitive Growth*

Modelling", Architectural Design, 76 Issue 2 March/April, 12-17.

Hensel, M., (2006b), "*Towards Self-Organisational and Multiple-Performance Capacity in Architecture*", Architectural Design, 76 Issue 2 March/April, 5-11.

Hight, C., and Hardt M., (2006), "*Designing Commonsplaces: Riffing with Michael Hardt on the Multitude and Collective Intelligence*", Architectural Design, 76 Issue 5 September/October, 70-73.

Hookway, B., and Perry, C., (2006), "*Responsive Systems – Appliance Architectures*", Architectural Design, 76 Issue 5 September/October, 74-79.

Jones, J. Christopher, Design Methods in Architecture, "*The state-of-the-art in design methods*", Architectural Association Paper no: 4, Ed. Geoffrey Broadbent, Anthony Ward, 1969

Keller, E., and Leita, C., (2006), "*Agent Intellects: Pattern as a Form of Thought*", Architectural Design, 76 Issue 5 September/October, 11-17.

Kennon, K., (2006), "*Does Collaboration Work?*", Architectural Design, 76 Issue 5 September/October, 50-53.

Kolarevic, B., "*Digital Morphogenesis*", "Architecture in Digital Age: Design and Manufacture" Spon Press, London, 2003

Lobel, J., (2006) "*The Milgo Experiment: An Interview with Haresh Lalvani*", Architectural Design, 76 Issue 4 July/August, 52-61.

Mayr, E. "*One long Argument: Charles Darwin and the Genesis of Modern Evolutionary Thought*", Cambridge (Ma), Harvard University Press, 1991

Mayr, E. "*Evolution and the diversity of life*", Harvard University Press, 1976.

McCullough, M., (2006), "*20 Years of Scripted Space*", Architectural Design, 76 Issue 4 July/August, 12-15.

Menges, A., (2006), "*Manufacturing Diversity*", Architectural Design, 76 Issue 2 March/April, 70-77.

Menges, A., (2006), "*Polymorphism*", Architectural Design, 76 Issue 2 March/April, 79-87.

Menges, A., "*Instrumental Geometry*", Architectural Design, 76 Issue 2 March/April, 43-53.

Morel, P., (2006), "*Computational Intelligence: The Grid as a Post-Human Network*", Architectural Design, 76 Issue 5 September/October, 100-103.

Ousterhout J., (1998) "*Scripting: Higher Level Programming for the 21st Century*", IEEE Computer

Raychman, John, "*The Deleuze Connections*" MIT Press, 2001

Reas, C., (2006), "*Process/Drawing*", Architectural Design, 76 Issue 4 July/August, 26-33.

Reas, C., "*The Language of Computers.*" Creative Code. John Maeda, Ed. Thames & Hudson, London, 2004

Rocker, M. I., (2006), "*Calculus-Based Form: An Interview with Greg Lynn*", Architectural Design, 76 Issue 4 July/August, 89-95.

Rocker, M. I., (2006), "*When Code Matters*", Architectural Design, 76 Issue 4 July/August,

16-25.

Salomon, D., (2006), “*Strength in Numbers*”, Architectural Design, 76 Issue 5 September/October, 55-57.

Shelden, D. R., (2006), “*Tectonics, Economics and the Reconfiguration of Practice: The Case for Process Change by Digital Means* Dennis R Shelden”, Architectural Design, 76 Issue 4 July/August, 82-87.

Silver, M., (2006), “*Codes, Eros and Craft: An Interview with Evan Douglass*”, Architectural Design, 76 Issue 4 July/August, 62-71.

Silver, M., (2006), “*Building Without Drawings: Automason Ver 1.0*”, Architectural Design, 76 Issue 4 July/August, 46-51.

Small, D., and Rothenberg, J., (2006), “*Design Research on Responsive Display Prototypes: Integrating Sensing, Information and Computation Technologies*”, Architectural Design, 76 Issue 5 September/October, 46-49.

Sprecher, A., Ahrens, C., and Neuman, E., (2006), “*Critical Practice: Protocol for a Fused Technology*”, Architectural Design, 76 Issue 5 September/October, 31-35.

Stathopoulos, N., Weinstock, M (2006) “*Advanced Simulation in Design*”, Architectural Design, 76 Issue 2 March/April, 54-59.

Steele, B., (2006), “*The AADRL: Design, Collaboration and Convergence*”, Architectural Design, 76 Issue 5 September/October, 58-63.

Tierney, T., (2006), “*Collective Cognition: Neural Fabrics and Social Software*”, Architectural Design, 76 Issue 5 September/October, 37-45.

Verebes, T., (2006), “*Associative Practices in the Management of Complexity*”, Architectural Design, 76 Issue 5 September/October, 65-69.

Weinstock, M., (2006), “*Self-Organisation and Material Constructions*”, Architectural Design, 76 Issue 2 March/April, 34-41.

Weinstock, M., (2006), “*Self-Organisation and the Structural Dynamics of Plants*”, Architectural Design, 76 Issue 2 March/April, 26-33.

Williams, C., “*Design by Algorithm*”, “*Digital Tectonics*”, United Kingdom: Wiley Academy, 2004

Wolfram, S., (2006), “*How Do Simple Programs Behave?*”, Architectural Design, 76 Issue 4 July/August, 34-37.

İNTERNET KAYNAKLARI

[1] www.tdk.gov.tr

[2] <http://wikipedia.org/>

[3] http://www.fep.up.pt/conferencias/EAEPE2007/Papers%20and%20abstracts_CD/Liagouras.pdf (*Socio-economic evolution and Darwinism in Thorstein Veblen: A post-Marxist appraisal* (George Liagouras, Department of Financial and Management Engineering, University of the Aegean, Greece))

- [4]<http://www.termbank.net/psychology/7182.html>
- [5]http://bio.research.ucsc.edu/~barrylab/classes/animal_behavior/HISTORY.HTM
- [6]<http://worldmake.blogspot.com/2006/04/summary-of-de-landas-lecture-on.html>
- [7]<http://www.goldsmiths.ac.uk/history/news-events/niamh-deleuze-paper.doc>
- [8]<http://www.talkorigins.org/faqs/evolphil/species.html>
- [9]http://www.berlage-institute.nl/03_postgraduate/PhDs/PhD.P.Trummer.html (*Peter Trummer; Population Thinking in Architecture (working title)*)
- [10]www.netmatematik.com/netmatematik/bolum4.html
- [11] <http://sci.ege.edu.tr/~mat/yazi/mobius.html>
- [12] <http://www.roboeducators.org/downloads/prog%20inst/Algorithmic%20Think.doc>
- [13] <http://www.bitstorm.org/gameoflife/>

EKLER

- Ek 1 3dsMAX'te Yazılan “Script”in Kodu
- Ek 2 Atölye Çalışmasındaki Ürünlerden Örnekler
- Ek 3 Atölye Çalışmasındaki Yorumlar

Ek 1 3dsMAX'te Yazılan "Script" in Kodu

```
-- // ALP ÇILTIK
-- // YTU - BOM - 07/08 FALL

-- // integers

c1 = 1; c2 = 1; c3 = 1; c4 = 1;
c5 = 1; c6 = 1; c7 = 1; c8 = 1;
--
splnsys = 10 ; j = 0 ; k = 0 ; i = 0; s=0; bxn=10; num=1

--
--Manipulators of Focus Points
ADDC1X = 0; ADDC4X = 0; ADDC5X = 0; ADDC8X = 0
ADDC1Y = 0; ADDC4Y = 0; ADDC5Y = 0; ADDC8Y = 0
ADDC1Z = 0; ADDC4Z = 0; ADDC5Z = 0; ADDC8Z = 0

--Manipulators of Other Points
ADDC2X = 0.5; ADDC3X = 0.1; ADDC6X = 0.1; ADDC7X = 0.1
ADDC2Y = 0.1; ADDC3Y = 0.1; ADDC6Y = 0.1; ADDC7Y = 0.1
ADDC2Z = 0.1; ADDC3Z = 0.1; ADDC6Z = 0.1; ADDC7Z = 0.1
--
--focus points
c1x = 0; c1y = 0; c1z = 0
c4x = 0; c4y = 0; c4z = 0
c5x = 0; c5y = 0; c5z = 0
c8x = 0; c8y = 0; c8z = 0

--spline points
c2x = 0; c2y = 0; c2z = 0
c3x = 0; c3y = 0; c3z = 0
c6x = 0; c6y = 0; c6z = 0
c7x = 0; c7y = 0; c7z = 0

--
thck = 0.1

--

rollout Pick_box1 "Kutu Seç 1"

(
    fn box_filt1 obj = classof obj == Box
    pickbutton chooseit "1. Kutu" width:140 filter:box_filt1
```

```

on chooseit picked obj do
(
    if obj != undefined do
    (
        obj.wirecolor = red
        obj.name = "BX1"
        chooseit.text = obj.name
        select $BX1
        $.lengthsegs = 5
        $.widthsegs = 5
        $.heightsegs = 5

        convertToMesh $
        clearselection()
    )
)
)

rollout Pick_box2 "Kutu Seç 2"

(
    fn box_filt2 obj = classof obj == Box
    pickbutton chooseit "2. Kutu" width:140 filter:box_filt2
    on chooseit picked obj do
    (
        if obj != undefined do
        (
            obj.wirecolor = red
            obj.name = "BX2"
            chooseit.text = obj.name

            select $BX2
            $.lengthsegs = 5
            $.widthsegs = 5
            $.heightsegs = 5

            convertToMesh $
            clearselection()
        )
    )
)

rollout Pick_box3 "Kutu Seç 3"

(
    fn box_filt3 obj = classof obj == Box

```

```

pickbutton chooseit "3. Kutu" width:140 filter:box_filt3
on chooseit picked obj do
(
    if obj != undefined do
    (
        obj.wirecolor = red
        obj.name = "BX3"
        chooseit.text = obj.name

        select $BX3
        $.lengthsegs = 5
        $.widthsegs = 5
        $.heightsegs = 5

        convertToMesh $
        clearselection()
    )
)
)

rollout Pick_box4 "Kutu Seç 4"

(
    fn box_filt4 obj = classof obj == Box
    pickbutton chooseit "4. Kutu" width:140 filter:box_filt4
    on chooseit picked obj do
    (
        if obj != undefined do
        (
            obj.wirecolor = red
            obj.name = "BX4"
            chooseit.text = obj.name

            select $BX4
            $.lengthsegs = 5
            $.widthsegs = 5
            $.heightsegs = 5

            convertToMesh $
            clearselection()
        )
    )
)

rollout Menu1 "Spline Sayisi"
(
    spinner splnsysS "Spline Sayisi:" range:[0,100,15] type:#integer

```

```

on splnsysS changed value do splnsys = value

    Label Labelinfo "Dikkat! Yüksek degerler yavas calisir" align:#center
)
rollout Menu2 "Odak Noktasi Koordinatlari"
(
    Label Labelinfo "Odak Noktasi 1" align:#center

    spinner C1XS "Odak 1 X:" range:[-500,500,0] type:#integer
    on C1XS changed value do C1X = value
    spinner C1YS "Odak 1 Y:" range:[-500,500,0] type:#integer
    on C1YS changed value do C1Y = value
    spinner C1ZS "Odak 1 Z:" range:[-500,500,0] type:#integer
    on C1ZS changed value do C1Z = value

    Label Labelinfo2 "Odak Noktasi 2 " align:#center
    spinner C4XS "Odak 2 X:" range:[-500,500,0] type:#integer
    on C4XS changed value do C4X = value
    spinner C4YS "Odak 2 Y:" range:[-500,500,0] type:#integer
    on C4YS changed value do C4Y = value
    spinner C4ZS "Odak 2 Z:" range:[-500,500,0] type:#integer
    on C4ZS changed value do C4Z = value

    Label Labelinfo3 "Odak Noktasi 3" align:#center
    spinner C5XS "Odak 3 X:" range:[-500,500,0] type:#integer
    on C5XS changed value do C5X = value
    spinner C5YS "Odak 3 Y:" range:[-500,500,0] type:#integer
    on C5YS changed value do C5Y = value
    spinner C5ZS "Odak 3 Z:" range:[-500,500,0] type:#integer
    on C5ZS changed value do C5Z = value

    Label Labelinfo4 "Odak Noktasi 4" align:#center
    spinner C8XS "Odak 4 X:" range:[-500,500,0] type:#integer
    on C8XS changed value do C8X = value
    spinner C8YS "Odak 4 Y:" range:[-500,500,0] type:#integer
    on C8YS changed value do C8Y = value
    spinner C8ZS "Odak 4 Z:" range:[-500,500,0] type:#integer
    on C8ZS changed value do C8Z = value
)
rollout Menu3 "Odak Noktasi Manipulatoru"
(
    Label Labelinfo "Manipulator 1" align:#center

    Label Labelinfo2 "ODAK noktaları X Manipulasyonu" align:#center
    spinner ADDC1XS "Odak 1 Pos. X:" range:[-40,40,0] type:#float
    on ADDC1XS changed value do ADDC1X = value
    spinner ADDC4XS "Odak 2 Pos. X:" range:[-40,40,0] type:#float

```

```

on ADDC4XS changed value do ADDC4X = value
spinner ADDC5XS "Odak 3 Pos. X:" range:[-40,40,0] type:#float
on ADDC5XS changed value do ADDC5X = value
spinner ADDC8XS "Odak 4 Pos. X:" range:[-40,40,0] type:#float
on ADDC8XS changed value do ADDC8X = value

```

```

Label Labelinfo3 "ODAK noktaları Y Manipulasyonu " align:#center

```

```

spinner ADDC1YS "Odak 1 Pos. Y:" range:[-40,40,0] type:#float
on ADDC1YS changed value do ADDC1Y = value
spinner ADDC4YS "Odak 2 Pos. Y:" range:[-40,40,0] type:#float
on ADDC4YS changed value do ADDC4Y = value
spinner ADDC5YS "Odak 3 Pos. Y:" range:[-40,40,0] type:#float
on ADDC5YS changed value do ADDC5Y = value
spinner ADDC8YS "Odak 4 Pos. Y:" range:[-40,40,0] type:#float
on ADDC8YS changed value do ADDC8Y = value

```

```

Label Labelinfo4 "ODAK noktaları Z Manipulasyonu " align:#center

```

```

spinner ADDC1ZS "Odak 1 Pos. Z:" range:[-40,40,0] type:#float
on ADDC1ZS changed value do ADDC1Z = value
spinner ADDC4ZS "Odak 1 Pos. Z:" range:[-40,40,0] type:#float
on ADDC4ZS changed value do ADDC4Z = value
spinner ADDC5ZS "Odak 1 Pos. Z:" range:[-40,40,0] type:#float
on ADDC5ZS changed value do ADDC5Z = value
spinner ADDC8ZS "Odak 1 Pos. Z:" range:[-40,40,0] type:#float
on ADDC8ZS changed value do ADDC8Z = value

```

```

)
rollout Menu4 "Diğer Noktaların Manipulasyonu"
(

```

```

Label Labelinfo "Manipulator 2" align:#center
Label Labelinfo4 "Manipulation of Spline Coordinates Z " align:#center

```

```

spinner ADDC2ZS "1. Spline Pos. Z:" range:[-40,40,0] type:#float
on ADDC2ZS changed value do ADDC2Z = value
spinner ADDC3ZS "2. Spline Pos. Z:" range:[-40,40,0] type:#float
on ADDC3ZS changed value do ADDC3Z = value
spinner ADDC6ZS "3. Spline Pos. Z:" range:[-40,40,0] type:#float
on ADDC6ZS changed value do ADDC6Z = value
spinner ADDC7ZS "4. Spline Pos. Z:" range:[-40,40,0] type:#float
on ADDC7ZS changed value do ADDC7Z = value

```

```

)
rollout Stuff "Render"
(

```

```

Label Labelinfo "Renderable Spline Thickness" align:#center
spinner thckS "Thickness:" range:[-3,3,0.06] type:#float

```

```

    on thckS changed value do thck = value

)

rollout INFO "INFO"
(
    Label LabelVersion "Nov 02 2007" align:#center
    Label LabelAuthor "Alp Ciltik" align:#center
    Button Contact "Contact: alp@ciltik.com" align:#center
    On Contact pressed do
    (
        assetBrowser.open()
        assetBrowser.gotoURL "mailto:alp@ciltik.com"
    )
)

-- // functions
fn drawLine point1 point2 point3 point4 point5 =
(
    aLine = SplineShape pos:point1 steps:12
    addNewSpline aLine --bir numalari egri
    addKnot aLine 1 #smooth #curve point1
    addKnot aLine 1 #smooth #curve point2
    addKnot aLine 1 #smooth #curve point3
    addKnot aLine 1 #smooth #curve point4
    addKnot aLine 1 #smooth #curve point5
    updateshape aLine --must
    aLine.wirecolor = color 228 214 153
    aLine.Renderable == true
    aLine.thickness = thck
)

fn drawLine point5 point6 point7 point8 =
(
    bLine = SplineShape pos:point5 steps:12
    addNewSpline bLine
    addKnot bLine 1 #smooth #curve point5
    addKnot bLine 1 #smooth #curve point6
    addKnot bLine 1 #smooth #curve point7
    addKnot bLine 1 #smooth #curve point8
    updateshape bLine --must
    bLine.Renderable == true
    bLine.thickness = thck
    bLine.wirecolor = color 228 214 153
)

```

```

rollout PSH "Bastirt"
(
    button Push "Lets Dance" width:240 height:60
    on Push pressed do

-- // program loop

    for k=1 to splnsys do
    (
        c1 = [c1x,c1y,c1z]
        c2 = getVert $BX1 (random 1 110)
        c3 = getVert $BX2 (random 1 110)
        c4 = [c4x,c4y,c4z]
        c5 = [c5x,c5y,c5z]
        c6 = getVert $BX3 (random 1 110)
        c7 = getVert $BX4 (random 1 110)
        c8 = [c8x,c8y,c8z]

        for i=1 to splnsys do
        (

-- // edit points "manipulasyon spline 1 drawing"
            -- x +
            c1.x = ADDC1X + c1.x
            c2.x = ADDC2X + c2.x
            c3.x = ADDC3X + c3.x
            c4.x = ADDC4X + c4.x

            -- y +
            c1.y = ADDC1Y + c1.y
            c2.y = ADDC2Y + c2.y
            c3.y = ADDC3Y + c3.y
            c4.y = ADDC4Y + c4.y

            --z +
            c1.z = ADDC1Z + c1.z
            c2.z = ADDC2Z + c2.z
            c3.z = ADDC3Z + c3.z
            c4.z = ADDC4Z + c4.z

-- // DRAW

            drawLine c1 c2 c3 c4

-- // edit points "manipulasyon spline 2 drawing"
            -- x +

```

```

c5.x = ADDC5X + c5.x
c6.x = ADDC6X + c6.x
c7.x = ADDC7X + c7.x
c8.x = ADDC8X + c8.x
    -- y +
c5.y = ADDC5Y + c5.y
c6.y = ADDC6Y + c6.y
c7.y = ADDC7Y + c7.y
c8.y = ADDC8Y + c8.y
    --z +
c5.z = ADDC5Y + c5.z
c6.z = ADDC6Z + c6.z
c7.z = ADDC7Z + c7.z
c8.z = ADDC8Y + c8.z

-- // DRAW

                                drawLine c5 c6 c7 c8
                                )
                                )
                                )
rollout TMZ "Temizle"
(
    button Push "Cut Throat" width:240 height:20
    on Push pressed do

                                delete $shape*
                                )

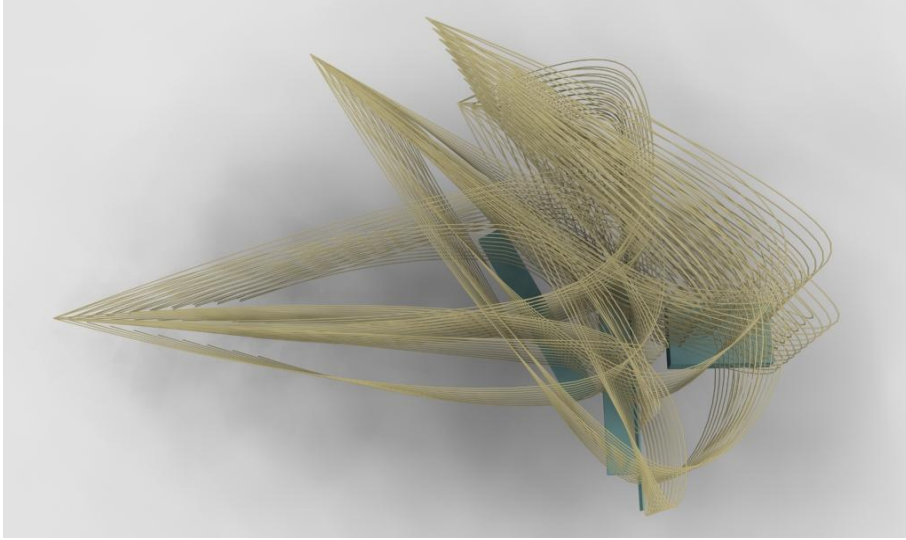
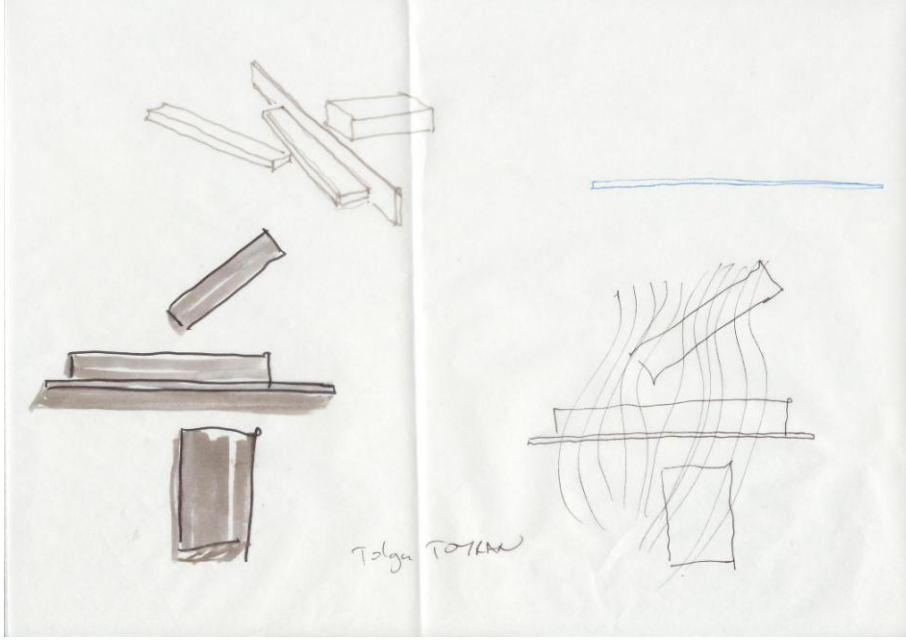
MENU = NewRolloutFloater "Dancing On Boxes" 270 600
AddRollout Pick_box1 MENU rolledup:off
AddRollout Pick_box2 MENU rolledup:off
AddRollout Pick_box3 MENU rolledup:off
AddRollout Pick_box4 MENU rolledup:off
AddRollout Menu1 MENU rolledUp:on
AddRollout Menu2 MENU rolledUp:on
AddRollout Menu3 MENU rolledup:on
AddRollout Menu4 MENU rolledup:on
AddRollout PSH MENU rolledup:off
AddRollout TMZ MENU rolledup:off
AddRollout Stuff MENU rolledUp:on
AddRollout INFO MENU rolledUp:on

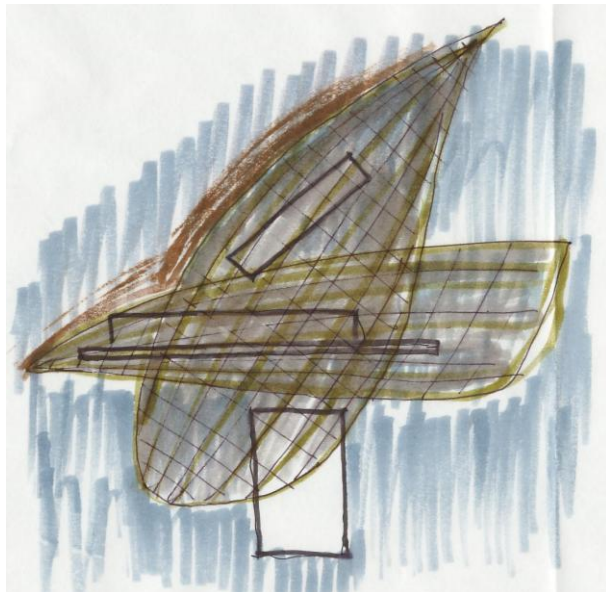
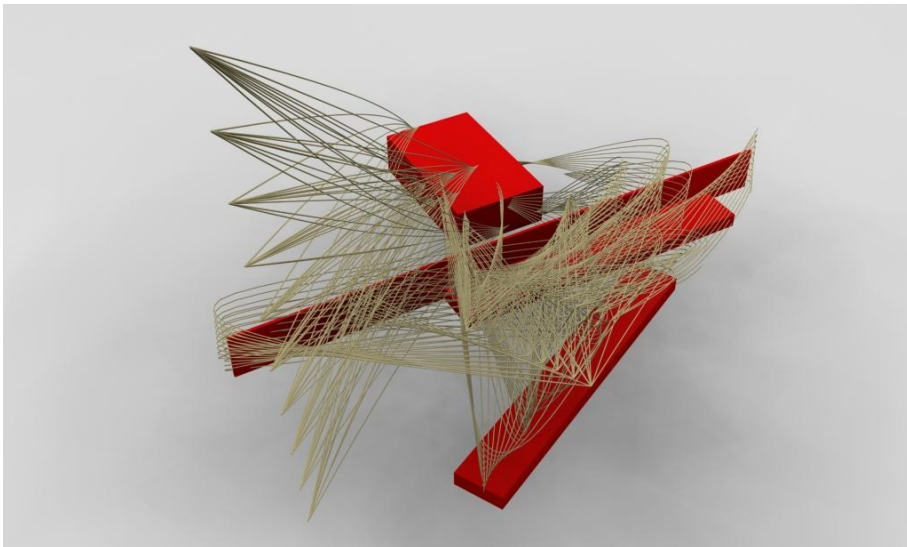
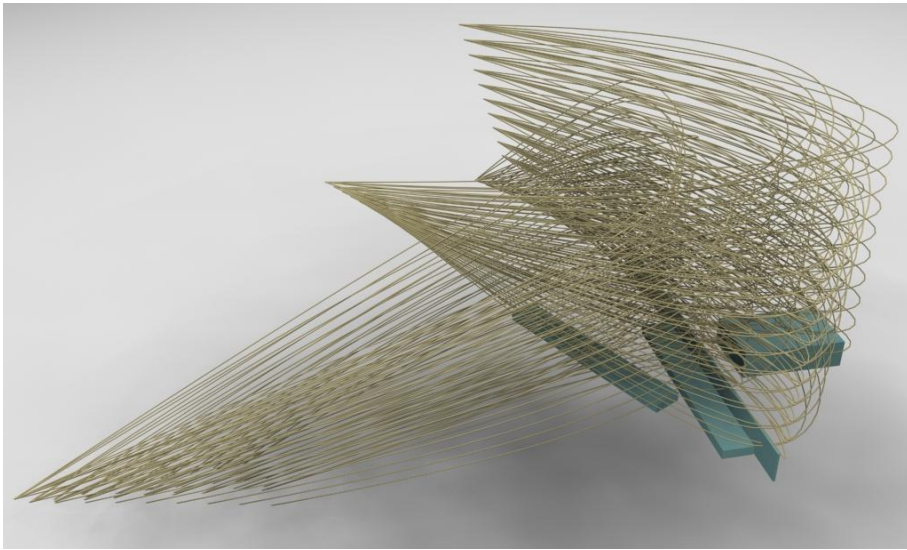
-- // finish

```

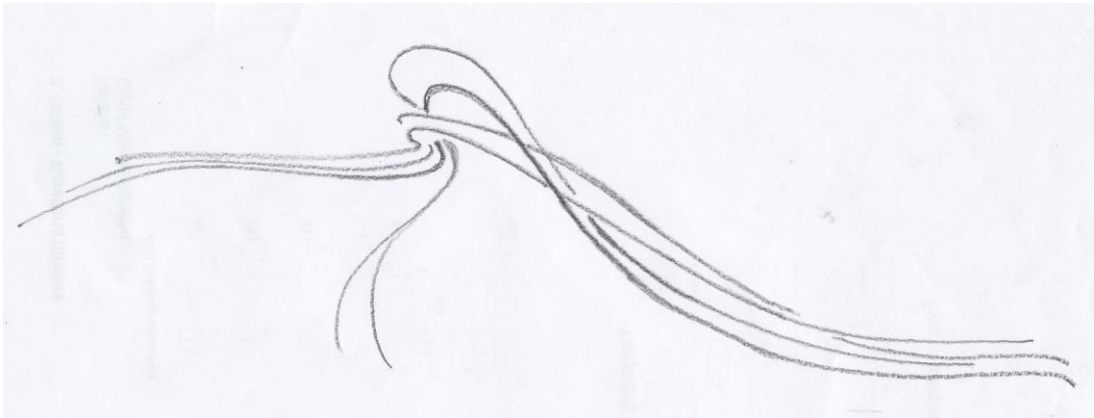
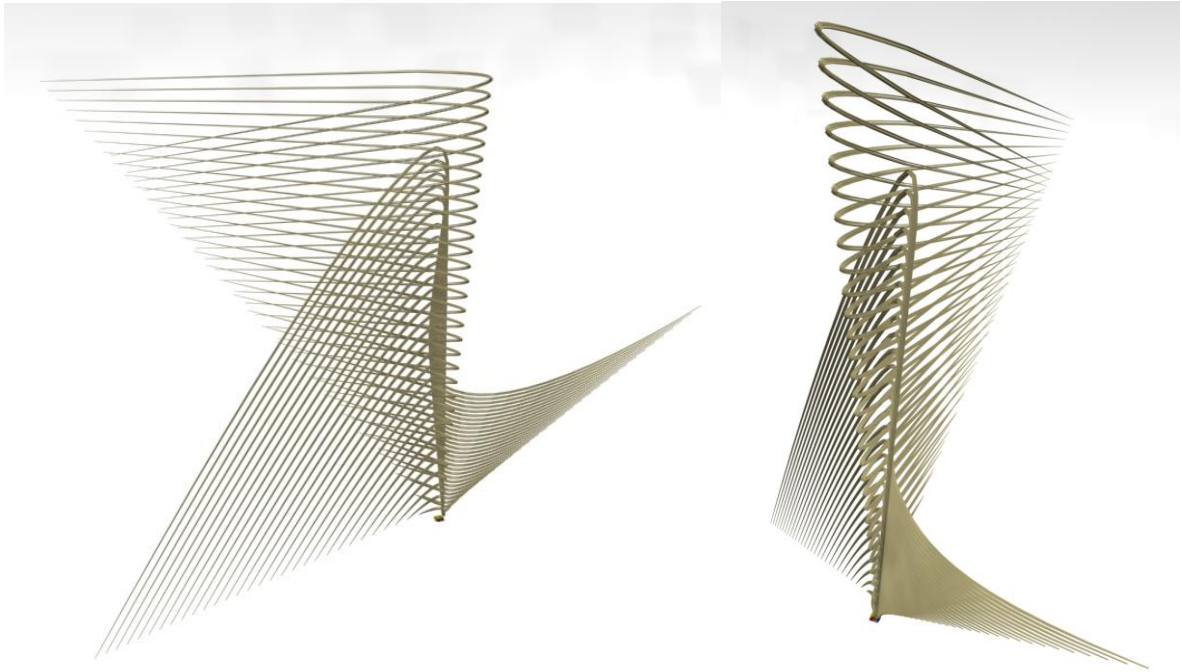
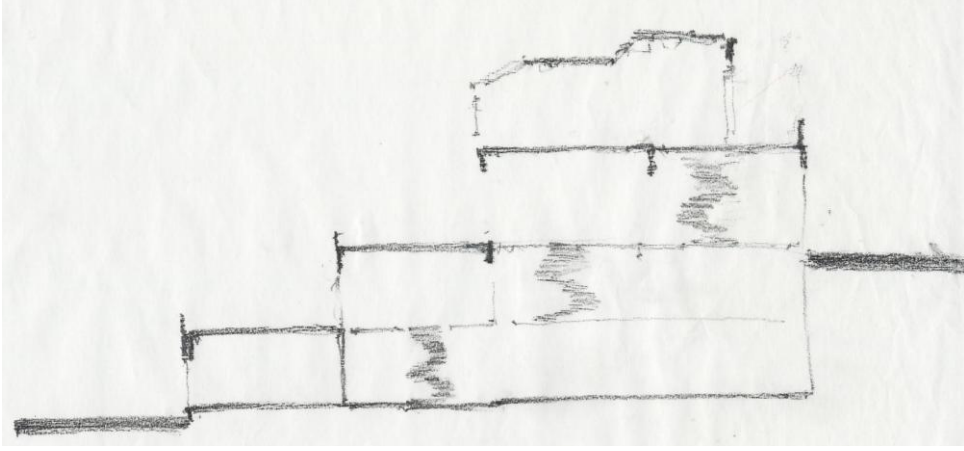
Ek 2 Atölye Çalışmasındaki Ürünlerden örnekler

1. Katılımcı

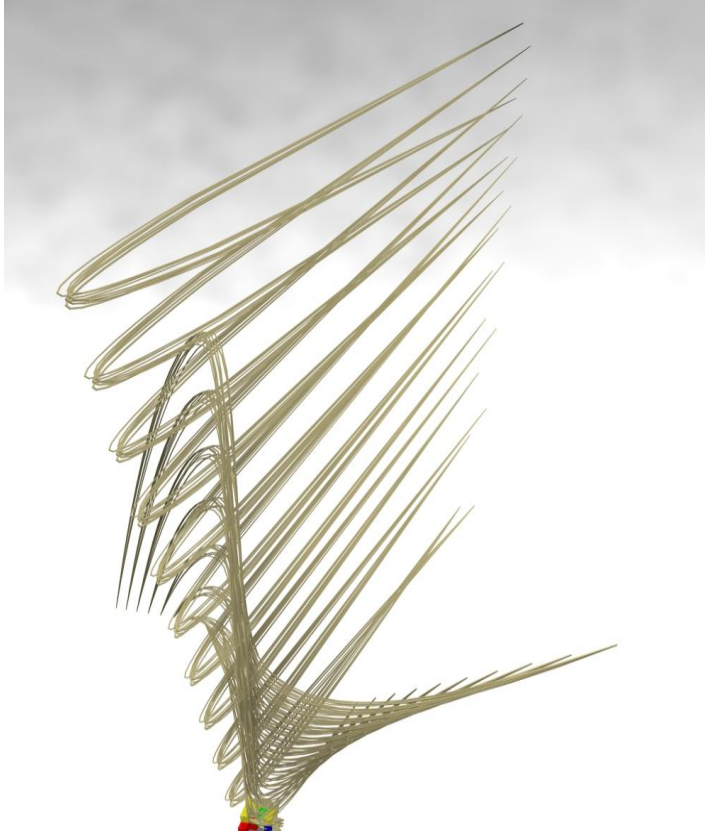
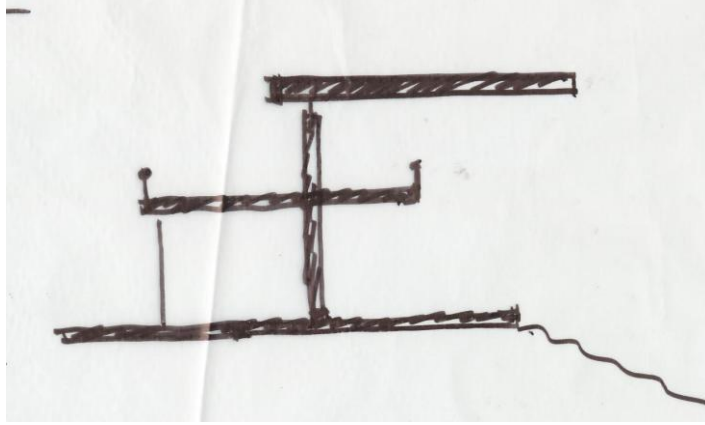


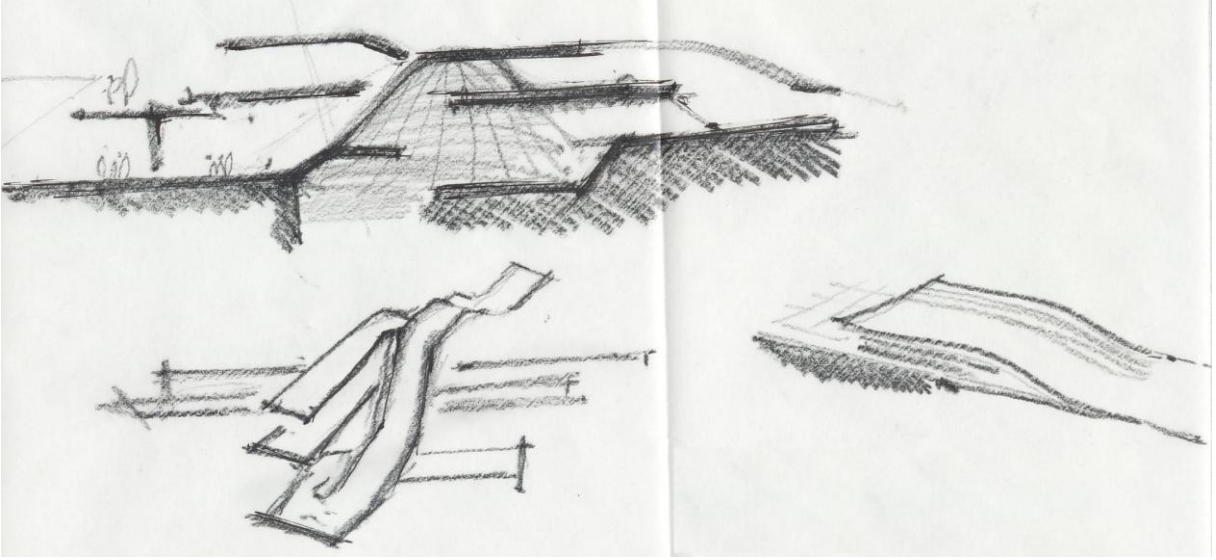


2. Katılımcı

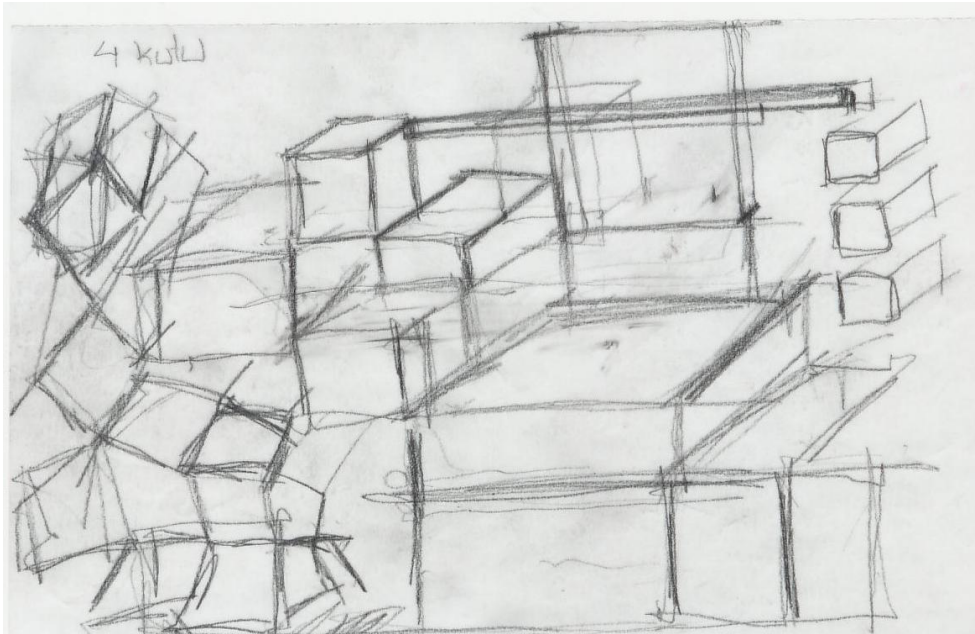


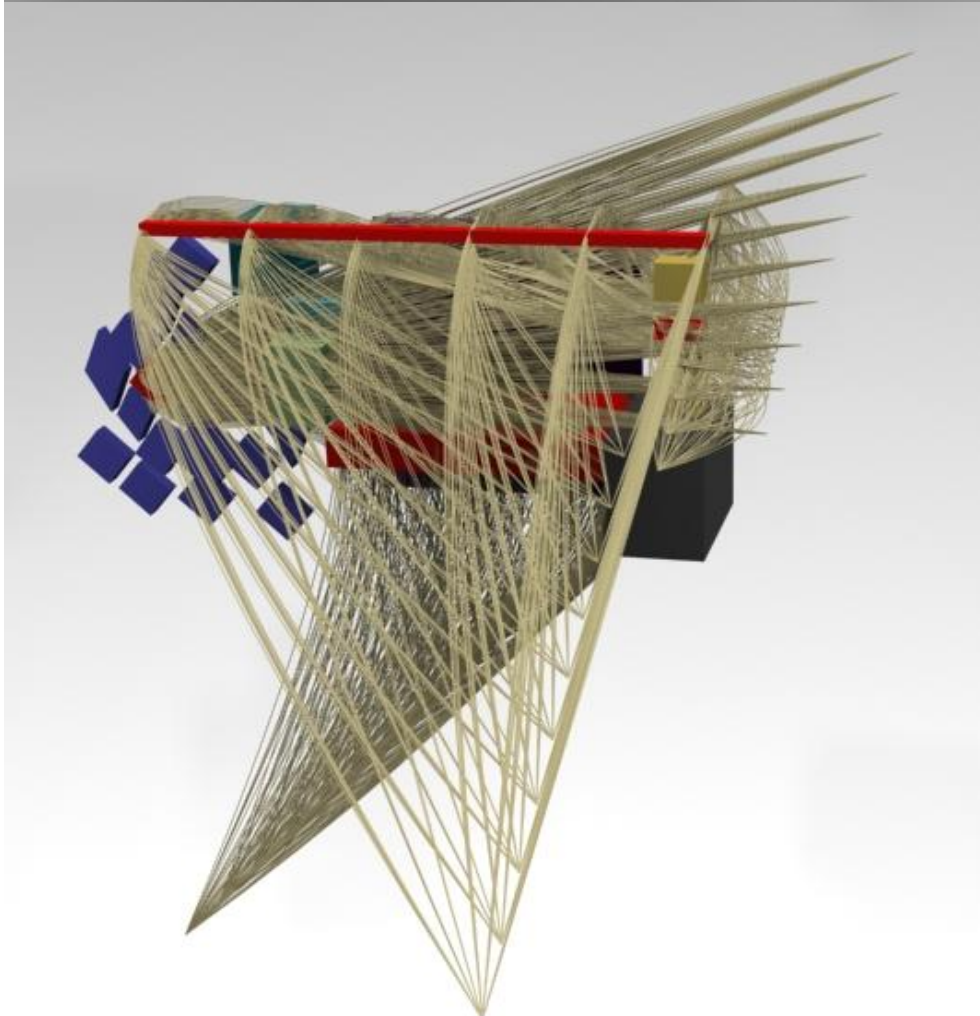
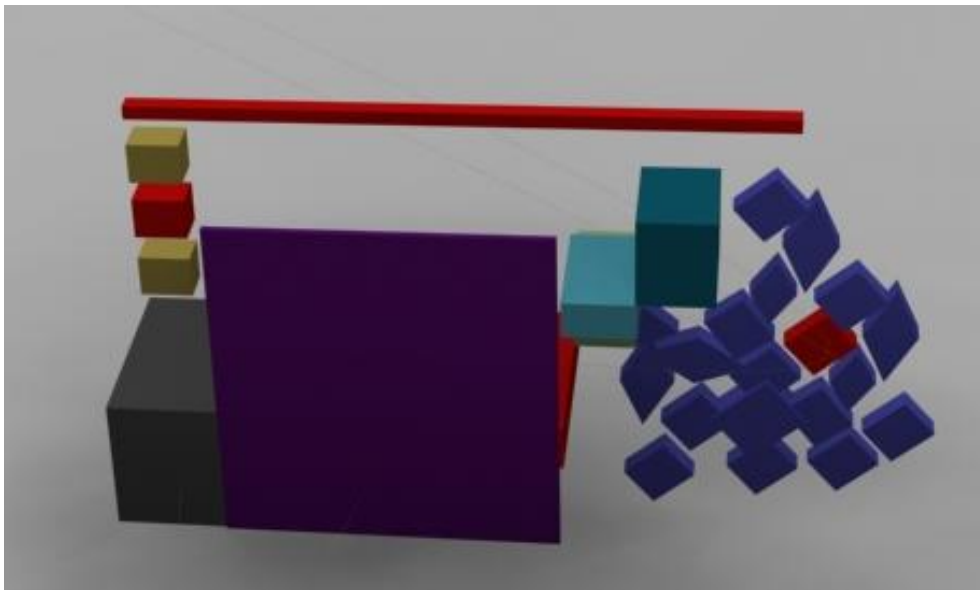
3. Katılımcı

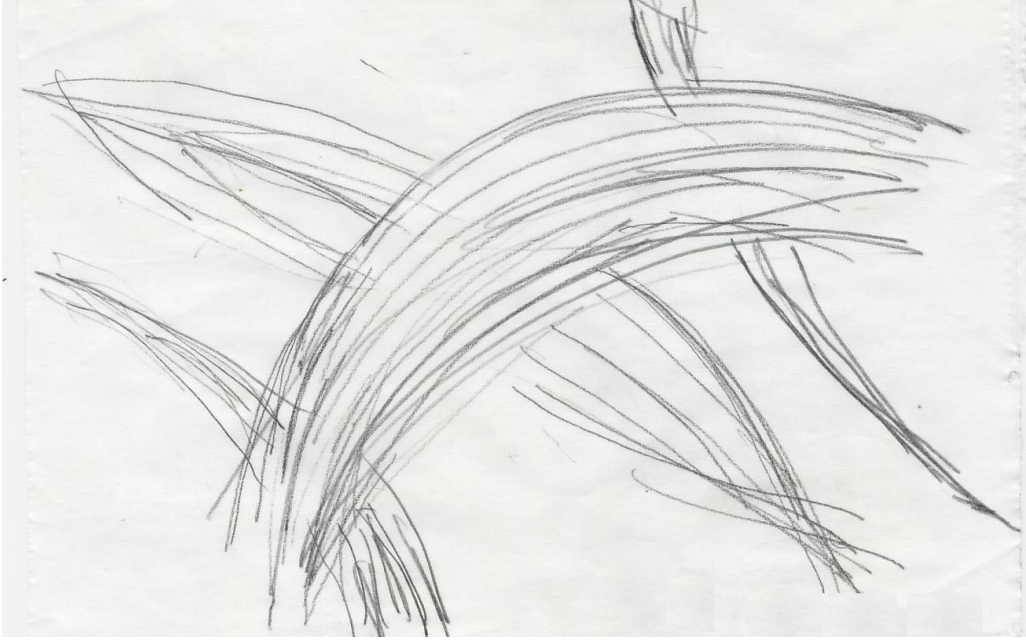




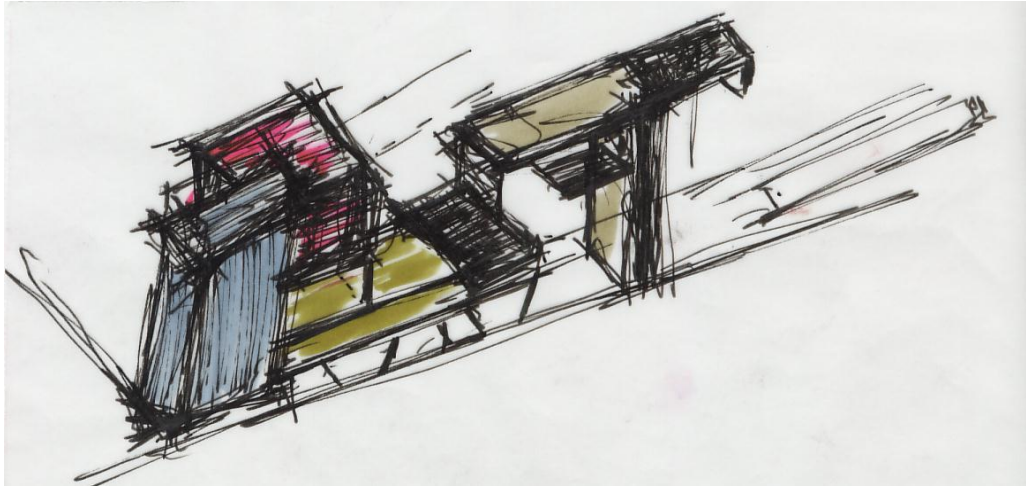
4. Katılımcı

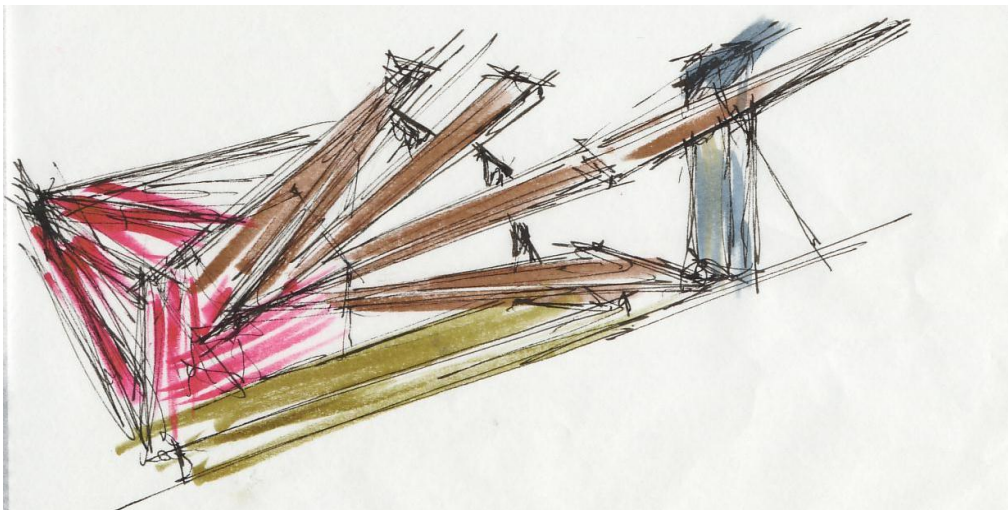






5. Katılımcı





Ek 3 Atölye Çalışmasındaki Yorumlar

Sorular

1. Aşama (ilk eskiz)

- A. Genel olarak tasarım başlangıcında yaptığınız eskiz sırasında ne gibi genel kavram ve girdiler üzerinde duruyorsunuz?
- B. İlk eskiz bölümünde eskiz yaparken sizi etkilediğini inandığınız genel veya size özel kavramlar nelerdir?

2. Aşama (program kullanımı)

- A. Programı kullanırken çıkan ürünleri tahmin edilebilir buluyor musunuz?
- B. Algoritmayı kullanırken zorlandınız mı? Fikirleriniz nelerdir?
- C. Algoritma yazımı mimarlar adına daha kolay olsa idi ilk tasarım aşamasında programlama dili kullanır mıydınız? Fikirleriniz nelerdir?

3. Aşama (ikinci eskiz)

- A. İkinci eskiz bölümünde eskiz yaparken sizi etkilediğini inandığınız genel veya size özel kavramlar nelerdir?
- B. İkinci eskiz süresinde kullandığınız programın etkisi oldu mu? Olduysa ikinci eskiz aşaması sırasında düşünme sürecinizi ne yönde etkiledi?

Yorumlar

1. Katılımcı

1A - Genel olarak tasarım başlangıcında çevresel verilerden ve fonksiyonun getirdiği zorunluluklardan yararlanırım. Tasarım çok geniş bir uzay olduğu için, çerçeveyi daraltmak adına bu iki faktörü kullanırım.

1B - Ürünün ölçeğiyle ve konusuyla olan ilişkisi benim için önemlidir. Kısaca uyum veya tezatlık üzerine yoğunlaşırım.

2A - İlk ürünleri tahmin edemedim. Ancak “script”i ve programı kurcaladıkça; elde edilen ürünleri tahmin edilebilir buldum. Manipulatörlerin çalışma mantığını kavradıktan sonra istediğim ürünü ortaya koymak için manipültörlerle oynamaya başladım.

2B - Hayır, zorlanmadım

2C - İlk tasarım aşamasında genel kriterler konulması ve tasarım uzayını sınırlandırması açısından yararlı olabilir. Yalnız konu derinleştikçe ve tasarım ilerledikçe; daha çok kural, daha çok kısıtlayıcı ve daha çok faktör etkilemeye başlayacaktır. Bundan dolayı kuralları koymak ve programlama diline aktarmak tasarımı elle yapmaktan çok daha uzun vakit alacaktır.

3A - Sanırım en önemli özellik oran oldu benim için.

3B - Bu kütlelerle çalışırken ve programı kullandıktan sonra programdaki bazı beğendiğim ürünlerden de esinlenerek ikinci eskizi yaptım. Bana bu programı yazan kişinin istediği ürünle kendi ürünüm arasında ortak bir ürün bulmam için yol gösterdi.

2. Katılımcı

1A - Tasarım problemi içinde verilen girdileri değerlendirip, öncelikle boyutlar, oranlar, kalemin akışını belirler.

1B - Araziye uyum, doğaya ve insana yakınlık

2A - Ürünler ilk etapta kolaylıkla tahmin edilmese de altındaki analitik fikri anlayınca ürünlerin tahmin edilmesi kolaylaşıyor.

2B - Ürünlere çok hızlı erişiliyor. Çok zorlanmadım. Bir bakış açısı; bakış yönü farklı olan için anlaşılması zor olacaktır. Fakat istediklerime ulaşmak için zorlandığım anlar oldu. Yazılıma müdahale etmek isterdim

2C - Kolay olsaydı fikrimi hangisi hızlı ve beni rahat ifade ederse o yolu kullanırdım. Bu yolu algılayan ve dillendirmesini beceren için çok güçlü bir araç olacaktır.

3A - Algoritmanın altında yatan mantık bana değil yazana ait olduğundan formal kavramlarımı etkiledi.

3B - Kullandığımız yazılım kullandığım çizgilerin üstüne tekrar düşünmemi sağladı. Biçimlenmesi ile etkileşime girdim

3. Katılımcı

1A - Tasarım çalışmalarında yapacağım çalışmanın ne ölçekte bir ürün çıkaracağı konusu ilk önemli kriterimdir. Çünkü ürünün ölçeği nasıl bir teknik kullanmam gerektiğini ve ne kadar ileri gidebileceğimi belirler. (Bir masa ile şehir ölçeğinde bir tasarım arasındaki farklara binaen) Tasarıma başlarken benim için en önemli veri çevresel faktörlerdir.

1B - Ürünün niteliğine bağlı olarak “algı” ve “ışık” benim için önemli verilerdi.

2A - Programa çok hakim olmadığım için, çıkan ürünün tahmin edilir olup olmadığıyla ilgili bir fikrim yok.

2B - Zorlanmadım. Sonuçlar gerçekten şaşırtıcı ve futuristikti.

2C - Kesinlikle kullanırdım. Çünkü geometrisi farklı işlerde, bu ürünlerin maketi herhangi bir yolla yapılabilir. Fakat teknik çizim haline getirmek oldukça zor. Ayrıca bu program eskiz aşamasında oldukça faydalı olurdu.

3A - “Boşluk” ve “sınır” benim verilerimdi.

3B - Fazlasıyla etkiledi. Programın ritmik ve çizgisel yapısı heyecan vericiydi.

4. Katılımcı

1A - Fonksiyon, yönelim, girişler, yatay-düşey sirkülasyonlar.

1B - Vaziyet planında yerleşim, yönelim ve gabariler ilk eskizde etkilendiğim konulardır.

2A - Hayır

2B - -

2C - Zorlandım. Eğrilerin koordinatlarının değişim kriterlerini kavramakta zorlandım. Manipülatör, manipülasyon, Let's Dance yerine Türkçe kullanımının zorunluluğu göz önünde bulundurulmalı. Ekran üzerinde nokta yakalamak ve bu noktaları değiştirebilmek gerekliliği.

3A - Bütünlük kaygısı

3B - Olmadı.

5. Katılımcı

1A - Arsa, iklim, yön, manzara gibi somut kavramlar üzerinde duruyorum. Çünkü mimari o yapıda yaşayacak olan insanların yaşamını, ruh durumunu doğrudan etkiler. Dolayısıyla doğrudan insanı etkileyeceğini düşündüğüm bu kavramlar bir eskize başlarkenki önceliklerimdir.

1B - Genel kavramları A şıkında açıkladım diyebilirim. Özel koşullar olarak da örneğin - benim yaşadığım mekanın olmazsa olmazı olarak sayabileceğim- güneşi ve mutfak-yemek odası, banyo-yatak odası gibi mekanların birbirleriyle bağlantılı olması gerektiğidir.

2A - Bazı ürünler tahmin edilebilir. Çünkü zaten belli doneler standart olarak programa yüklenmiştir.

2B - Zorlanmadım ama hangi parametre neyi etkiliyor, daha iyi anlamak gerekli. Bu da zaman v programı kullanımının sıklığıyla ilgili. Sonuçlar ilginç ve iyiydi.

2C - Programı daha iyi algıladığımda kullanabilirdim. Ama her tasarım için kullanılabilecek bir program olduğunu düşünmüyorum.

3A - “Sonsuzluk” kavramı ve fikir anlamında insan beynini farklı yönlere yöneltmesi anlamında etkilendim.

3B - Kesinlikle oldu. Fikir anlamında daha önce aklıma gelmeyecek tasarımları daha çok düşünebilirim programdan sonra.

6. Katılımcı

1A - Tasarım probleminde yer alan unsurlar eskiz için ön girdileri oluşturur. Bu problematik çerçevesinde sezgisel olarak hareket ederim.

1B - İlk eskiz çalışmasında verilen 3 kutu çalışmanın genel prensibini oluşturdu. Bu kutuları rasyonel bir geometrik organizasyonla mekan oluşturulmaya çalışıldı.

2A - Hayır edemiyorum. Programlama diline çok hakim olmadığım için çıkan girdilere rastlantısal olarak ulaşıyorum.

2B - Zorlandığımı söyleye bilirim. Tasarımlarda sezgisel hareket ettiğimden dolayı, analitik olan bu kurguya çok hakim olamadım.

2C - Eğer eskize aktarmayı düşündüğüm fikri bilgisayara kolaylıkla tanıttırabilirsem kullanabilirim.

3A - Programlama sonucunda elde ettiğim sonuç ürünler ikinci eskizi daha amorf formlar üzerinde çalıştım. Genel tasarım problemi aynıydı ancak çizgiler daha serbestti diyebilirim.

3B - Programlamadan çıkan ürünlerin yeni eskize etkisi oldu. Yeni form arayışlarına yöneldiğimden bir başka deyişle ilk eskizde var olan rasyonel çizginin kırılmaya çalışıldığından bahsedebilirim.

7. Katılımcı

1A - Tasarıma başlangıç eskizinde verilen tasarım problemi çerçevesinde içgüdüsel bir tasarım hareketinden bahsedebilirim. Bu bakış açısı mimari tasarım problemine göre değişiklik gösterirken, mimari eğitimin başlangıcından bu güne kadar olan algıların, deneyimlerin ve farklı arayışların etkisinden söz edilebilir.

1B - Bu eskiz için belirlenen ana kriter dört kutu aracılığı ile bir mekan tanımı olmuştur. Araştırılan yeni mekanı geliştirirken hedeflenen, bu basit geometrilerin birbiri ile ilişkisi, oranları ve bu geometrileri ne şekilde bir araya getirilebilir sorusuna cevap arayan bir form araştırmasıdır.

2A - Programa çok hakim olamadığımdan ilk çıkan ürünleri ön göremiyordum ancak denemeler sonunda, görmek istediğim sonuca benzer sonuçlar ürettim.

2B - Mimaride matematiksel yargılarla tasarım yapmaya alışık olunmadığından bu algoritmik kurgu içinde sonuca ulaşmakta ilk etapta zorlandım. Programın işleyişini algıladıktan sonra daha efektif sonuçlara ulaştım.

2C - Algoritmanın kullanımında tasarım probleminin kapsamı önem kazanıyor. Genel çözümler için tasarım ilk aşamasından son ürüne kadar bu kurgu içinde tanımlanabilir ancak spesifik çözümlerde bu kurgunun tasarıma girdi oluşturan bir form çalışması kapsamından uygulamaya yönelik bir sonuç ürüne götürmenin zor olduğunu düşünüyorum. Ancak tasarımcının ön göremediği formları üretmek için önemli bir adım olduğunu söyleyebilirim.

3A - Yapılan çalışmayı programla tanıttıktan sonra elde edilen yeni girdiler, yapılan ikinci eskiz çalışmasında farklı formsal deformasyonları araştırmamı sağladı. Yapılan ikinci eskizde basit geometrik ilişkilerle geliştirilmiş olan form çalışması, geometrik ilişkiler bazında nasıl farklılaştırılabilir sorusuna cevap arayışı bağlamında geliştirildi.

3B - Programın etkisinin olduğundan bahsedebilirim. İlk eskizde rasyonel bir çözüm üretilmişken, ikinci eskiz çalışmasını, bu rasyonel çizginin hangi yönde kırılabilceği sorusuna cevap arayan bir form çalışması olarak değerlendirilebilir. Bu amaç doğrultusunda program sonucunda üretilen yeni ürünlerin, eskizin gelişiminde bir araç olarak kullanıldığından söz etmek gerekir.

8. Katılımcı

1A - Projeden gelen istekler ve alanın verdiği veriler doğrultusunda eskize başlayıp kendi mimari beğenimlerimi kullanmaya çalışırım

1B - Lineer bir formdan oluşan parçalı bir kompleks yapmayı düşündüm

2A - Program kısa süreli kullanımda çok etkili ürünler veremeyebiliyor. Ancak programın kullanımına alışılması ile çıkan ürünler konsept ürün anlamında kullanılabilir

2B - Max içindeki programcığı kullanım açısından zorluğu yok fakat mantığı anlamak için uzun süreli kullanım veya bilen birinin anlatması gerekiyor.

2C - Programlama dillerinin çıkan ürün açısından kısa zamanda kullanılabilir veya kullanılamaz olmasına karşın fazla ürün ortaya çıkarması açısından kullanmanın çok olumlu olacağını düşünüyorum.

3A - Programdan gelen ürün ile parçacıklı yapıları bir çatı örtüsü olarak kullanmayı düşündüm.

3B - Bana göre yazılan program yüzey kaplaması açısından kullanılabilir. Günümüzde mimarinin bilgisayar ile bu denli dirsek temasında bulunması beraberinde bu tarz programların yakın gelecekte mimari bürolarda kullanılacağını gösteriyor. Her büro programı kullanacağı projeye ve kullanma amacına göre tekrar yazması koşulu ile çok verimli olacağını düşünüyorum.

ÖZGEÇMİŞ

Doğum tarihi 28.05.1980

Doğum yeri İzmir

Lise 1994–1996 Kocaeli Körfez Fen Lisesi
1996–1997 Karşıyaka Lisesi

Lisans 1998–2004 İzmir Yüksek Teknoloji Enstitüsü
Mimarlık Fakültesi, Mimarlık Bölümü

Yüksek Lisans 2004- Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Mimarlık Anabilim Dalı,
Bilgisayar Ortamında Mimarlık Programı

Çalıştığı kurumlar

2002–2004	Azan Kozbe Ltd Şti.	(İzmir)
2005–2007	Pidar Proje A.Ş.	(İstanbul)
2007-	Yüzbaşıoğlu İnşaat A.Ş.	(İstanbul)