

# MİMARLIK EĞİTİMİNDE ALGORİTMA: STÜDYO UYGULAMALARI

**Birgül ÇOLAKOĞLU ve Tuğrul YAZAR**

Bilgisayar Ortamında Tasarım Bilim Dalı, Mimarlık Fakültesi, Yıldız Teknik Üniversitesi, Yıldız, İstanbul  
[bigi\\_c3@yahoo.com](mailto:bigi_c3@yahoo.com), [turuly@yahoo.com](mailto:turuly@yahoo.com)

(Geliş/Received: 18.07.2006; Kabul/Accepted:17.08.2007)

## ÖZET

Sayısal teknolojilerin gelişmesiyle birlikte mimarlık alanında yapılan araştırmalar, tasarımda hesaplamalı yöntemlere odaklanmaktadır. Hesaplamalı yöntemler, formel olguları farklı teknikler ve metodlar ile yeniden gündeme getirmiştir. Algoritmik mantıkla çalışan araçlar, bu araştırmaların odak noktasıdır. Mimarlıkta bilgisayarın yaygın kullanımına rağmen, algoritmanın mimari tasarımda kullanımı sınırlıdır.

Bu çalışma, “Hesaplamalı Tasarımın Tasarım ve Mimarlık Eğitimine Entegrasyonu” araştırması kapsamında yürütülen, “Tasarımı Tasarlamak” adlı deneysel atölye çalışmasını incelemektedir. YTÜ Mimarlık Fakültesi Bilgisayar Ortamında Mimarlık Yüksek Lisans programında yürütülen atölyenin amacı, algoritmik tasarım mantığının mimari tasarım sürecine entegrasyonunu irdelemektir.

Bu makale üç bölümde hazırlanmıştır. İlk bölümde tasarımın formel dili ve algoritmik tasarım irdelenmiş, ikinci bölümde, algoritmik tasarımın yöntem olarak kullanıldığı atölye çalışması ve örnekleri incelenmiş, üçüncü bölümde atölye çalışmasının ve değişen tasarım paradigmalarının değerlendirilmesi yapılmıştır.

**Anahtar Kelimeler:** Mimarlık eğitimi, algoritma, parametrik tasarım, nesne-tabanlı programlama.

## ALGORITHM IN ARCHITECTURAL EDUCATION: STUDIO WORKS

### ABSTRACT

Recent research in architecture have focused on to understand computational methods of formal exploration and expression. Computational mechanisms have been used to readdress formal issues using new techniques and methods. Computational tools that operate on algorithmic logic are central in these explorations. Despite the proliferation of computers in architecture today, use of algorithms in architectural design is limited.

This paper describes “Designing the Design” studio work, developed within the scope of “The integration of Computational Design into Design and Architectural Education” research project. The aim of the studio is to explore the use of algorithms in architectural design process.

The paper is organized in three parts, the first part elaborates formal language and algorithmic approach to design. The second describes the studio work in which algorithmic logic is integrated into design process and the third evaluates the studio work and discusses the changing paradigms in practice and education of design.

**Keywords:** Architectural education, algorithm, parametric design, object-oriented programming.

### 1. GİRİŞ (INTRODUCTION)

Günümüz sayısal teknolojileri hem tasarımcı hem de tasarım eğitimcilerini yeni tasarım yöntemleri geliştirmeye ve formüle etmeye zorlamaktadır.

Sayısal teknolojilerin oluşturduğu ihtiyaçlar, fırsatlar ve süreçler son on yıl içerisinde dünya mimarlık okullarının ders programlarının yeniden yapılanmasında önemli rol oynamıştır [1]. Bunun sonucu olarak, mimarlık ders programlarına

“Algoritmik Tasarım,” “Türetici Tasarım” gibi yeni konuları inceleyen dersler eklenmiştir [2,3]. Uygulamada, özellikle tasarım konseptlerinin kavranması ve geliştirilmesinde, matematiksel bilgisayar algoritmaları önemli rol oynamıştır [4,5].

Bu çalışmada, sayısal tasarım araçlarının sunduğu olanaklar ile gelişen yeni eğitim paradigmasını irdelemek ve tasarım eğitimindeki olası entelektüel ve kuramsal yönlenmeleri araştırmak hedeflenmiştir.

## 2. TASARIMIN FORMAL DİLİ (FORMAL LANGUAGE OF DESIGN)

Birçok farklı bağlamda (bilim, tasarım, hukuk, dilbilim vb.) formel dilden<sup>1</sup> söz etmek mümkündür. Bilgisayar destekli tasarım araçları formel sistemlerdir ve formel dil kullanırlar. Bilgi, formel dil yardımı ile bilgisayara aktarılır ve kullanılabilir hale getirilir. Bu sistemler, tasarımcıyı soyutlanmış ve kullanılabilir hale getirilmiş bilgi bütünleri ile çalışmaya yönlendirdikleri için “geometrik farkındalık” niteliğinin ve algoritmik düşünce yapısının gelişmesine katkıda bulunurlar. Günümüzde tasarımcının bu yeni tasarım diline hakim olabilmesi için bu dili oluşturan analitik geometri, algoritma ve kompozisyon kurallarını kavraması gerekmektedir [6].

### 2.1. Geometri (Geometry)

Tasarımın soyut dili geometri üzerine kuruludur. Bilgisayar destekli tasarım araçları, tasarımın geometrik elemanlarını, bu elemanlar arasındaki her türlü ilişkiyi ve dönüşümü (transformasyonu) tasarımcıdan talep ederler. Tasarımcı yeni tasarım yöntemlerinde ancak bu talebin farkında olması ve buna cevap verecek şekilde kendini geliştirmesi ölçüsünde başarılı olabilmektedir.

### 2.2. Kompozisyon (Composition)

Sayısal tasarım sistemlerinin toplu üretimde bireyselleştirme amaçlı kullanımı ve sayısal üretim yöntemlerinin buna imkan verecek şekilde entegre gelişimi tasarım bütünü oluşturur. Bu bağlamda, tasarım bütünü nasıl alt birimlere bölüneceği önem taşımaktadır. Erken tasarım sürecinde tasarım bütünü alt parçaları kavramsal yapıda incelenirken, üretime yönelik bir çalışmada alt bileşenler, malzeme ve üretim yöntemleri doğrultusunda kısıtlamalar içerebilir. Kompozisyon stratejilerinin belirlenmesi ve geliştirilmesi yeni tasarım dilinde önemli bir tasarım becerisi olarak ortaya çıkmaktadır.

<sup>1</sup> Formel dil: Matematik, mantık ve bilgisayar bilimlerinde “formel dil” sınırlı sayıda kelimenin (karakter, dizi) sınırlı bir alfabaden seçilerek ve belirli kurallar kullanılarak oluşturduğu kurgudur. Formel dil teorisi bilim kuramında, formel dil kuramı başlığı altında incelenir.

### 2.3. Algoritma (Algorithm)

Algoritma, bir problemin sınırlı sayıda adımla çözüm sürecini ifade eder. Sistematik ve rasyonel düşünce örüntüleri gerektirmektedir. Mimari tasarım düşüncesi bu niteliği ile algoritmik düşünce ile örtüşmektedir. Geleneksel yöntemden farklı olarak, algoritmik düşünce yapısından yararlanıldığında tasarım problemlerine sayısal platformda çözüm(ler) üretme özelliği ön plana çıkmaktadır. Bu yöntem, tasarımcının ufku genişletmektedir.

Bir anlamda algoritmik düşüncenin altında “el-göz koordinasyonu” dışında üretilebilecek geometrileri keşfetme, diğer yanda geniş veri kümeleri kullanılarak üretilebilecek tasarımlarda “öngörülemleri” kontrol edebilme isteği yatmaktadır [7].

Bilgisayar destekli tasarım araçları, her bilgisayar yazılımında olduğu gibi bir algoritma kurgusu sayesinde çalışır ve kullanılırlar. Ancak bu yazılımlar genellikle araç olarak kullanılmak üzere kurgulandıkları için tasarım problemlerine çözüm olması muhtemel belirli nesne ve fonksiyonları hazır olarak sunmaktadırlar. Tasarımcıdan beklenen bu nesne ve fonksiyonları bir araya getirerek soruna çözüm üretmesidir. Oysa tasarımcı, yazılımın sunduğu nesne ve fonksiyonları değiştirmeye, veya kendisi nesne ve fonksiyonlar üretmeye ve kullanmaya başladığı zaman yazılımın algoritmik yapısını da çözümlenmeye başlar. Bu sürecin devamında tasarımcının tasarım problemine algoritmik düşünce yapısını kullanarak çözüm araması ve bu çözümü bilgisayar yardımıyla doğrudan doğruya hayata geçirmesi beklenir. Tasarımcı, kullandığı en yakın araç olan bilgisayarın algoritmik dili ile konuşmaya başladığı zaman, düşündüğü çözümü doğrudan doğruya görme ve farklı alternatifleri talep etme özgürlüğünü elde eder.

### 3. Tasarımı Tasarlamak (Designing the Design)

YTÜ Mimarlık Fakültesi Bilgisayar Ortamında Mimarlık Yüksek Lisans programında uygulanan “Tasarımı Tasarlamak” atölyesinin amacı, katılımcıları formel tasarım mantığı ve bunun getirdiği yeni tasarım dili ile tanıştırmaktır<sup>2</sup>. 2005-2006 öğretim yılı güz yarıyılında her hafta bir gün süresince devam eden atölye çalışmalarının yapısını “okuma”, “teknik bilgi” ve “parametrik tasarım uygulamaları” olmak üzere üç bölümde incelemek mümkündür.

<sup>2</sup> Atölyenin katılımcıları 15 kişilik yüksek lisans öğrenci grubudur. Gerekli teknik ve teorik destek aynı yüksek lisans programındaki Biçim Gramerleri (Y.Doç.Dr. Birgül Çolakoğlu) ve 3D Modelleme ve Animasyona Giriş (Öğr.Gör.Dr. Togan Tong) dersleri ile sağlanmıştır.

### 3.1. Okuma (Reading)

Atölyenin ilk bölümünde hedeflenen, katılımcıları tasarım dünyasında gerçekleştirmekte olan paradigma değişiminden haberdar kılmaktır. Bu bağlamda literatürde yayınlanmış örnekler incelenerek tasarımcı profilinde ve meslek pratiğinde gerçekleşen dönüşüm irdelenmiştir. Atölyenin genelinde, katılımcıların bu dönüşümden haberdar olmalarının yanında, bu dönüşümü bizzat deneyimlemelerini sağlamak hedeflenmiştir.

### 3.2. Teknik Bilgi (Technical Information)

Atölyenin ikinci bölümü, teorik bilginin pratiğe dönüştürülmesini amaçlamaktadır. Bu bölümde algoritmik tasarım mantığı ve tekniği üzerinde durulmuştur. Günümüzde Bilgisayar Destekli Tasarım (BDT) yazılımları, yeniden yapılanma ve güncellenme süreci içerisindedir. Söz konusu güncellemelerin önemli bir bölümünde hedeflenen, yazılımları daha esnek ve kullanıcı dostu kılmaktır. Bu amaçla yazılımlar içerisinde script<sup>3</sup> dilleri geliştirilmekte veya mevcut script yapıları güçlendirilmektedir. BDT yazılımları içerisindeki script dilleri (AutoLisp, Maya Encoding Language MEL, MaxScript, RhinoScript vb.) tasarımcılara, tasarım eğilimlerini sayısal ortamda ifade etme yöntemini sunmaktadır. Bu yöntem sayesinde, ekran üzerindeki arayüz yardımı ile ifade edilen durağan tasarımların yerini, ilişkilendirilmiş biçimsel ve mekansal tanımları içerisinde tüm parametreleri ve gelişim süreçleri kontrol edilebilen dinamik tasarımlar almaktadır. Bilgisayar bu sayede sadece görsel nitelikleri ile değil, algoritmik yapısı ile de tasarımcıya destek olmaktadır.

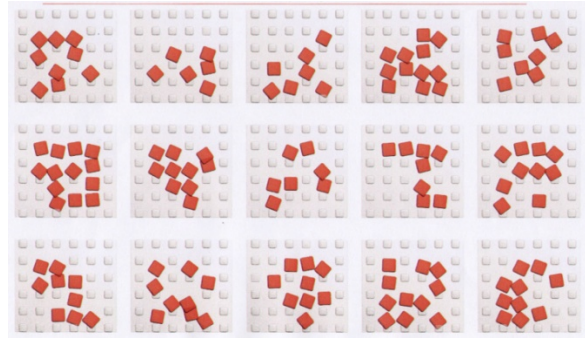
Atölye katılımcılarının bu yöntemi kullanabilmesi için bilgisayar programcısı olmaları gerekmemektedir. Ancak mimari tasarımda ve/veya temsilde kullandıkları araçların yapısındaki algoritma düzenlerini ve tasarım geometriyi kavramaları önemlidir. Atölyenin bu bölümünde katılımcılara maxscript<sup>4</sup> üzerinden söz konusu mantık ve geometri tanıtılmıştır.

<sup>3</sup>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 sözdizim kurallarına önem verilir. Script dillerinde ise uygulamayı hızlandırmak, tanımlanmış bütünleri bağlamayı kolaylaştırmak için sözdizim 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 öngörülmeyen başka bir durumdaki çözüm için kolaylıkla kullanılabilir [8]. Bu nitelikleri sayesinde script dilleri, diğer disiplinlerin bilgisayar programcılığı ile bağlarını kurabileceği en yakın köprü olma özelliğine sahiptirler.

<sup>4</sup>MaxScript'in uygulama alanı olarak seçilmesinin en önemli sebepleri arasında, yazılımın yaygın kullanımı, güçlü nesne yapısı ve yazılım hakkındaki mevcut kaynakların çokluğu gelmektedir.

Programlama dillerinin temel koşullama (if / then), tekrar etme (while / for) ve kullanıcı ile iletişime geçme (input / get) yöntemleri, diğer script dillerinde olduğu gibi maxscript'te de sadeleştirilerek kullanılmaktadır. Bilgisayar programcılığına ve script dillerine yabancı olan atölye katılımcılarına öncelikle yukarıdaki mantıksal ifadelerin söz dizim kuralları tanıtılmıştır.

Atölyenin ilk uygulamaları alıştırmaya niteliğindedir ve yazılım içerisinde mevcut olan parametrik biçim sınıflarını kullanma, biçimler arasında standart ilişkileri programlama ve belirli bir işlevi gerçekleştiren algoritma parçalarını (fonksiyonları) kurgulama konularını içermektedir. Şekil 1'de görülen örnek, konu ile ilgili yapılmış bir öğrenci uygulamasını göstermektedir. Bu uygulamada amaçlanan, mantıksal karşılaştırmaları kullanarak bilgisayarın koşulları test ediş biçimini, geometrik biçimlerin matematiksel olarak tanımlanmasını, biçimlerin transformasyonlarından yeni biçimler türetilmesini ve bütün bu işlemler için gerekli parametrik yapının kurgulanmasını irdelemek olmuştur.



Şekil 1. Script alıştırmaları (Scripting exercises)

Script dilinden faydalanabilmek için, kullanıcının yazılımın standart kullanımına da hakim olması esastır. Atölyenin bu bölümüne paralel olarak aynı programdaki diğer bir ders (3D Modelleme ve Animasyona Giriş) ile katılımcıların bu olası eksiği kapatılmıştır.

### 3.3. Parametrik Tasarım Uygulamaları (Parametric Design Exercises)

Atölye uygulamaları çeşitli nesne ailelerinin türetilmesini içermiştir. Katılımcılar için hem kullanılan araç hem de yöntem açısından yeni olan bu uygulamalar dört aşamalı olarak geliştirilmiştir.

**Analiz:** İlk aşama, tasarım nesnesini oluşturan alt bileşenlerin ve tasarımı etkileyecek olan temel parametrelerin keşfedilmesine dayanan analiz çalışmasıdır. Burada örnekler incelenerek kullanılacak tasarım elemanları belirlenmiştir.

**Sentez:** İkinci aşama, elde edilen verilerin tasarlanan nesne ailesini oluşturmak üzere sentezlenmesidir. Bu aşamada tasarımcı kullanacağı veya göz ardı edeceği

parametreleri ve alt bileşenleri seçer, algoritmik ilişkileri diagramlar ile tanımlar.

**Uygulama:** Üçüncü aşama, tasarım elemanlarının, elemanlar arasındaki ilişkilerin, parametrelerin ve sınırlamaların bilgisayara aktarılmasıdır.

**Test:** Son aşama, tanımlanmış tasarım algoritmalarını farklı parametreler ve kısıtlamalar ile çalıştırarak tasarım kompozisyonlarını deneme aşamasıdır.

### 3.3.1. Parametrik Mobilya Ailesi Tasarımı (Parametric Furniture Design)

Atölyenin ilk uygulaması bir mobilya ailesinin tasarımıdır. Katılımcılardan bir mobilya tasarımını parametrik olarak gerçekleştirmeleri ve sonuç ürün ailesini script dilinde oluşturmaları istenmiştir. Aşağıda bu çalışmanın sonuçlarından birisi anlatılmaktadır:

Birinci aşamada, mobilya ailesinin elemanları, genel alt biçimleri ve bu biçimler arası olası ilişkiler saptanmıştır –*analiz*-. Bu analiz çalışmasında mümkün olduğunca az parçanın bir araya farklı gelişlerinden en fazla sayıda kombinasyon elde etmek hedeflenmiştir. Hem mevcut mobilya modellerinden faydalanarak bunların biçimsel ve işlevsel olarak parçalanması ve yeniden bir araya getirilmesi, hem de yeni modüllerin tasarlanması yaklaşımları kullanılmıştır.

İkinci aşamada, biçimlerin alabileceği en çok / en az parametreler ve aralarındaki ilişkiler tanımlanmıştır –*sentez*-. Biçimler arası ilişkiler, biçimlerin genel bir araya geliş kurallarının, sonrasında da detaylarının belirlenmesi ile tanımlanmıştır.

Üçüncü aşamada, bu parametreler ürün ailesi elemanlarını oluşturmak için 3D Studio Max programının hazır arayüz nesnelere yararlanılarak “MobiDiz<sup>5</sup>” isimli bir arayüze aktarılmıştır –*uygulama*-.

Tasarım, stüdyo ortamındaki öğrenci sunumlarına ve tartışmalara göre şekillenmiştir. Sonuç ürün bir nesnenin tasarımı değil, bir nesne ailesinin tasarımıdır.

Şekil 2’de görülen arayüz yardımı ile tasarımcı, kurguladığı düzen içerisinde farklı sonuçlar elde edebilmektedir. Şekil 3’de bu uygulamanın sonuçlarından birisi görülmektedir –*test*-.

### 3.3.2. Parametrik Cephe Tasarımı (Parametric Facade Design)

Katılımcılardan farklı boyut ve işlevlere sahip binalardan oluşan bir yapı kompleksi için cephe ailesi alternatifleri üretmeleri istenmiştir. Aşağıda bu uygulamaya katılan altı proje grubundan üçünün çalışmaları incelenmiştir:



Şekil 2. Maxscript ile hazırlanmış “MobiDiz” arayüzü (Graphical User Interface of “MobiDiz” maxscript)



Şekil 3. Mobilya ailesi kombinasyonlarından birisi. (A furniture combination created by MobiDiz utility).

### Çalışma Grubu A (Workgroup A)

Bu grubun<sup>6</sup> çalışmasında mevcut arazide konumlanacak kütlelerin üzerini örten bir kabuk tasarımı amaçlanmıştır. Bunu yaparken alışılmış tasarım yöntemlerinden koparak bilgisayar destekli bir süreç kullanılmıştır. Yapı kütleleri, bir deneysel çalışma olan ve 3D Studio Max altında plug-in olarak çalışan ConGen’den [9] yararlanılarak, boyutları tasarımcı tarafından belirlenen minimum ve maximum noktalar arasında, bilgisayarda oluşturulmuştur –*analiz*-. Şekil 4’te ConGen ile elde edilen yapı blokları kompozisyonu görülmektedir.

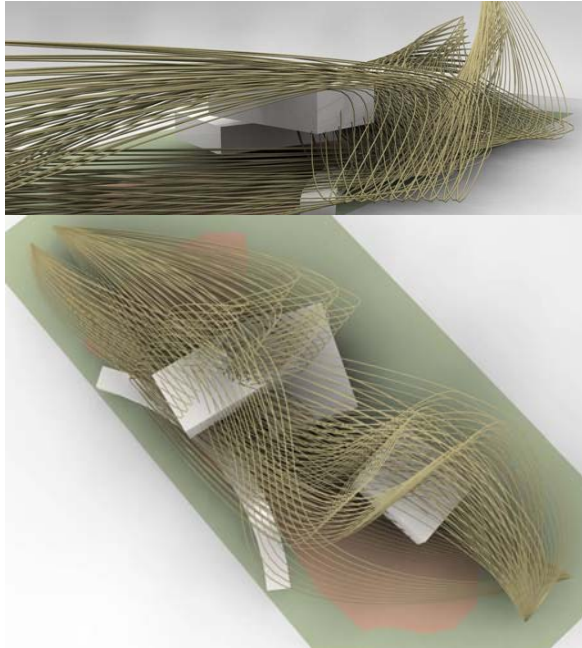
<sup>5</sup> Öğrenci çalışması: Y.Mimar Ayşegül Erin

<sup>6</sup> Öğrenci grubu: mimar Alp Çıltık, mimar Hüseyin Tok ve mimar Şirin Özbek.



Şekil 4. ConGen de üretilmiş yapı blokları kompozisyonu (Composition of building blocks developed by ConGen utility).

Çalışma grubu A'daki katılımcılar tarafından hazırlanan maxscript programı ile kütlelerin üzerine örten bir kabuk oluşturulması hedeflenmiş, tasarıma başlamadan önce gerekli parametreler, kütlelerin üst yüzeyleri ve dört odak noktası ile oluşturulan düzlemler gözden geçirilmiştir –*sentez*-. Tasarlanması amaçlanan örtüyü oluşturmak için iki odak noktasını ve seçilen iki kütlelerin üst yüzeylerine teğet olacak veya kesecek biçimde tanımlanan analitik denklemler kullanılmış, daha sonra bu denklemler, yazılan script ile manipüle edilerek olası farklı kabuk tasarımları üretilmiştir –*uygulama*-. Şekil 5' te yapılan bu üretmelerden örnekler görülmektedir.

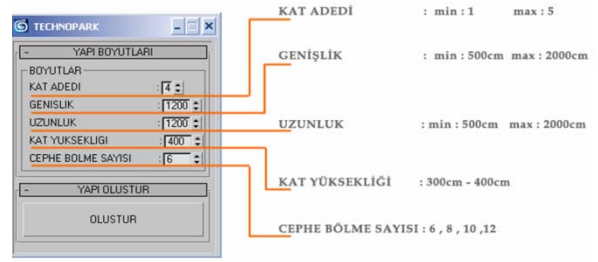


Şekil 5. Kabuk örtü üretmeleri(Example generations).

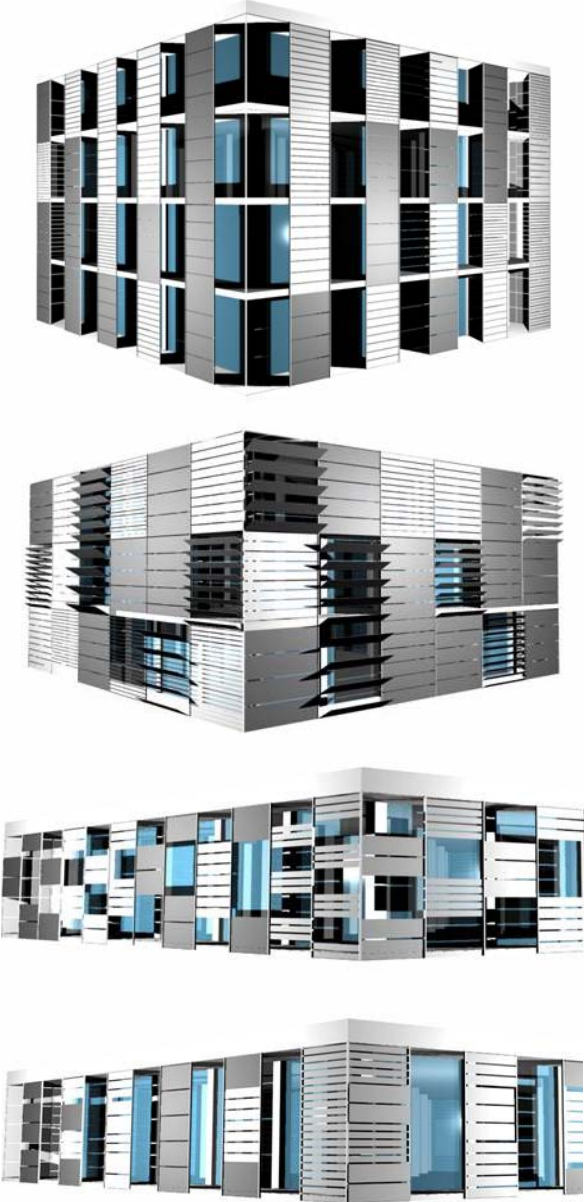
#### Çalışma Grubu B (Workgroup B)

A grubundan farklı olarak bu katılımcı grubu<sup>7</sup> mevcut yapı kütlelerinin saran iki boyutlu cephelere malzeme, dinamizm ve kompozisyon bağlamında yaklaşmıştır.

<sup>7</sup> Öğrenci grubu: mimar Gökhan Keskin, mimar Serhat Anıktar, mimar Sinem Öztürk ve mimar Can Akın.



Şekil 6. Cephe panelleri türeten arayüz(Interface of the parametric facade utility)



Şekil 7. Hareketli panellerden oluşan cephe fonksiyonunun farklı boyutlardaki yapılaraya uygulanması ve panellerin hareket biçimleri(Implementation of the moving panels to variable building dimensions).

Yapı bloklarını ayrı ayrı değerlendiren grubun amacı, farklı boyutlardaki binaların cephelerini kaplayan tek bir algoritma oluşturmaktır. Analiz aşamasında

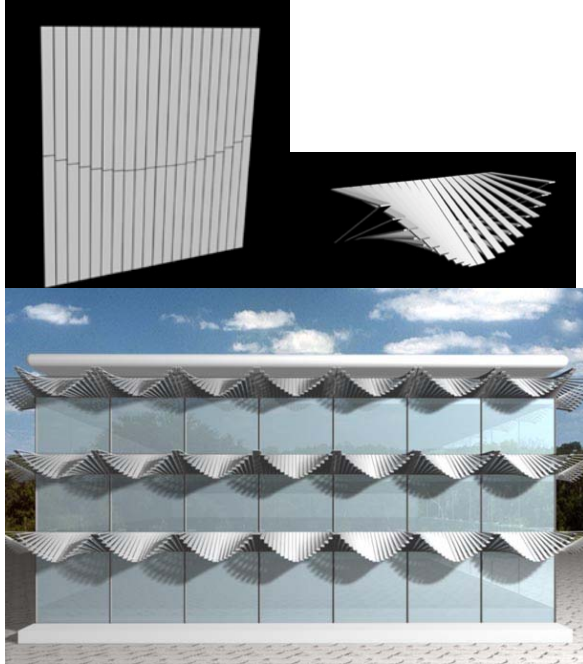
hareketli paneller ile dinamik cephe kurgularının oluşturulduğu örnekler teknik ve işlevsel açıdan değerlendirilmiştir. Sentez aşamasında ise bina cepheleri için gün ışığına göre hareket edebilecek aktif cephe panelleri öngörülmüştür. Uygulama aşamasında bina iskeleti oluşturulduktan sonra farklı bina boyutlarına uyum sağlayacak şekilde belirli cephe karakterleri parametrikleştirilmiş bu parametreler bir arayüz üzerinde toplanarak tasarım üzerinde hakimiyet sağlanmıştır.

Şekil 7'de sonuçları görülen bu çalışmanın ileri aşamalarında konuma ve saate göre gün ışığı tepkilerinin benzetimi öngörülmüştür.

#### Çalışma Grubu C (Workgroup C)

Cepheleri, B grubunda olduğu gibi kütlelerin yan yüzeylerini örtmek üzere kurgulayan bu grubun<sup>8</sup> yaklaşımındaki farklılık, mevcut bir mimari nesnenin parametrikleştirilmesi çabasında yatmaktadır. Santiago Calatrava'nın 1983 yılında Almanya'da gerçekleştirmiş olduğu ambar kapısı tasarımı, C grubunun analiz aşamasındaki çıkış noktasını oluşturmuştur. Sentez aşamasında söz konusu tasarımı yorumlayarak yeniden türetme denemiştir. Grup, proje raporunda probleme getirdikleri çözüm önerisini şu şekilde özetlemiştir:

*“Sistemi oluşturan alt ve üst paneller arasında eklem noktaları bulunmaktadır. Bu eklem noktalarının z düzlemindeki konumları, yapının bütününde bir parabol eğrisini ifade etmektedir. Problemi çözmeye*



**Şekil 8.** Hareketli modül fonksiyonu ve bir cephedeki uygulama önerisi (Dynamic modular structure and its implementation on a building facade).

<sup>8</sup> Öğrenci grubu: mimar Eda Türay ve mimar Yıldız Gergün.

*bu parabol eğrisinin denklemi bulunarak başlandı; “ $y=0.00091x^2-0.2696x+20$ ” Bu sayede kapalı ünite oluşturuldu. Açık ünite için ise yine bu denklemi kullanarak her panel için bir dönme (rotation) açısı üretildi. Panellerin tamamı denkleme bağlı olarak dönünce ünite açılmış oldu. Çalışmanın son aşamasında açık ve kapalı üniteleri biraraya getiren bir algoritma ve kullanıcı arayüzü oluşturularak 1,2,3 ve 4 katlı binaların cephelerine farklı alternatifler türetildi.”*

Hareketli, açılıp kapanabilen alt ve üst panellerden oluşan cephe elemanı sonuçları Şekil 8'de görülmektedir. Bu grubun çalışmaları için ileri aşamalarda eklem noktalarına dayalı gerçek zamanlı bir kinetik benzetim öngörülmüştür.

#### 4. SONUÇ (CONCLUSION)

Sayısal teknolojilerin sunduğu olanaklar ve fırsatlar tasarım uygulamasında ve eğitiminde paradigma değişimine neden olmaktadır. “Tasarımı Tasarlamak” atölye çalışması bu değişim sürecinde yeni tasarım yöntemlerinin tasarım eğitimi entegrasyonunu araştıran deneysel bir eğitim çalışması niteliği taşımaktadır. Bu çalışmada mimari programlar, tipolojiler, bina kodları ve mimari dilde varolan kural tabanlı mantığın form ve mekan tasarımında bir yöntem olarak kullanılması denemiştir.

Tasarımda tanımlı ve anlaşılabilir modellerin oluşturulması ve geliştirilmesi, algoritmik düşüncenin kullanımı ile doğrudan ilgilidir. Algoritmik tasarım, biçim, strüktür ve süreçleri inceleyen kavramsal bir çerçeve oluşturarak bilgisayar yazılımları içerisinde gömülü olan soyut tasarım uzayını görünenin ötesine taşımaktadır.

Sayısal teknolojilerin tasarım ve üretim aşamasında getirdiği yenilikler dünyasında mimarın yerini bulması için kendisine yabancı olan algoritmik form geliştirme ve tasarım süreçlerini tasarım yöntemlerine entegre etmesi gerekliliği hem tasarımcı hem de tasarım eğitimcileri arasında yoğun bir şekilde tartışılmaktadır. Bu bağlamda yapılan araştırma ve denemelerin insan yaratıcılığının yerini alabilecek çalışmalar olarak değerlendirilmemesi gerekir. Bu çalışmalar algılama veya yorumlama ile değil, insan aklının tasarım ifadesi için kullanabileceği yeni yöntemler ile ilişkilidir.

Geliştirilen bu yöntemler geleneksel tasarım yöntemlerinin yerini almayı değil onları desteklemeyi hedeflemektedir. Uygulama dünyasının gerçekleri ve rekabet ortamı uygulamacı tasarımcıyı bu yeni yöntemleri tasarım süreçlerine daha çabuk adapte etmeye zorlamaktadır. Mimarlık eğitiminin de bu değişime kendini adapte etmesi kaçınılmazdır.

**KAYNAKLAR (REFERENCES)**

1. Kvan, T., Mark, E., Oxman, R., ve Martins, B., “*Ditching the Dinosaurs: Redefining the role of Digital Media in Education*”, **International Journal of Design Computing**, No:7, 2004.
2. Knight, T., “*Shape grammars in education and practice: history and prospects*”, **International Journal of Design Computing**, No:2, 1999.
3. Terzidis, K., “*Algorithmic Design: A Paradigm Shift in Architecture?*”, **Education of Computer-aided Architectural Design in Europe, (eCAADe) 22. Konferans Bildirileri**, sf. 201-207, Kopenhag, 2004.
4. Balmond C., **Informal**, Prestel, Londra, 2002.
5. Kolarevic, B., **Architecture in the Digital Age, Design and Manufacturing**, Spoon Press, 2003.
6. Aish R., “*From Intuition to Precision*”, **Education of Computer-aided Architectural Design in Europe, (eCAADe) 23. Konferans Bildirileri**, sf. 10-14, Portekiz, 2005.
7. Kolarevic B., “*Designing and Manufacturing Architecture in the Digital Age*”, **Architectural Information Management 05, Design Process No: 3**, sf.117-123, 2005.
8. Ousterhout J., **Scripting: Higher Level Programming for the 21st Century**, *IEEE Computer*, 1998.
9. Çolakoğlu B. ve Dionyan S., “*A Parametric Form Generator: ConGen*”, **Education of Computer-aided Architectural Design in Europe, (eCAADe) 23. Konferans Bildirileri**, sf. 623-628, Portekiz, 2005.