

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YAPI BİLGİ MODELLEME TABANLI HESAPLAMALI TASARIM
YAKLAŞIMI İLE MİMARİ TASARIM İŞ AKIŞLARININ VERİMLİLİĞİNİN
SAĞLANMASI**

YÜKSEK LİSANS TEZİ

Ayşegül Özlem BAYRAKTAR SARI

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

HAZİRAN 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YAPI BİLGİ MODELLEME TABANLI HESAPLAMALI TASARIM
YAKLAŞIMI İLE MİMARİ TASARIM İŞ AKIŞLARININ VERİMLİLİĞİNİN
SAĞLANMASI**

YÜKSEK LİSANS TEZİ

**Ayşegül Özlem BAYRAKTAR SARI
(523141017)**

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Prof. Dr. Meryem Birgül ÇOLAKOĞLU

HAZİRAN 2018

İTÜ, Fen Bilimleri Enstitüsü'nün 523141017 numaralı Yüksek Lisans Öğrencisi Ayşegül Özlem BAYRAKTAR SARI, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “YAPI BİLGİ MODELLEME TABANLI HESAPLAMALI TASARIM YAKLAŞIMI İLE MİMARİ TASARIM İŞ AKIŞLARININ VERİMLİLİĞİNİN SAĞLANMASI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Meryem Birgül ÇOLAKOĞLU**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Salih OFLUOĞLU**
Mimar Sinan Güzel Sanatlar Üniversitesi

Doç. Dr. Hakan YAMAN
İstanbul Teknik Üniversitesi

Teslim Tarihi : **2 Mayıs 2018**
Savunma Tarihi : **1 Haziran 2018**





Aileme,



ÖNSÖZ

Tez hazırlama sürecinde akademik birikimi, tecrübesi ve kişiliğiyle beni en doğru biçimde yönlendiren, olumlu yaklaşımlarıyla tezin hazırlanmasına önemli katkılar sağlayan tez danışmanım Prof. Dr. M. Birgül Çolakoğlu'na, yüksek lisans eğitimim boyunca vizyonlarını paylaşarak mimarlığa farklı bir bakış açısı kazanmamı sağlayan İstanbul Teknik Üniversitesi Mimari Tasarımda Bilişim Programı öğretim üyelerine,

Hayatımın her anında desteklerini hissettiğim, her zaman arkamda olduklarını bildiğim, her düştüğümde elimden tutup kaldıran mutluluk sebeplerim sevgili annem ve sevgili babama, beni her zaman motive eden sevgili kardeşlerime, her zaman sevgi ve özveriyle yanımda olup beni cesaretlendiren sevgili eşime ve ikinci ailem olan sevgili Sarı Ailesi'ne, büyük çınarımız dedeme ve her hatırladığımda tebessüm ettiğim babaanneme ve diğer sevgili aile üyelerime,

Her zaman yanımda olan ve desteklerini esirgemeyen sevgili dostlarıma,

Sonsuz teşekkürler.

Haziran 2018

Ayşegül Özlem BAYRAKTAR SARI
(Mimar)



İÇİNDEKİLER

Sayfa

ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Tezin Amacı ve Kapsamı	1
1.2 Tezin Yöntemi.....	3
2. YAPI BİLGİ MODELLEME (YBM).....	5
2.1 YBM Temsil Seviyeleri	8
2.1.1 Projenin detay seviyesi.....	8
2.1.2 Modelleme boyutu	11
2.1.3 YBM olgunluk seviyeleri.....	13
2.2 YBM ve Proje Aşamaları	14
2.2.1 Ön tasarım aşamasında YBM kullanımı	15
2.2.2 Tasarım aşamasında YBM kullanımı.....	15
2.2.3 İnşaat aşamasında YBM kullanımı	16
2.2.4 Kullanım aşamasında YBM kullanımı.....	17
2.3 YBM Sisteminde Kullanılan Standart ve Formatlar	17
2.3.1 YBM Sınıflandırma Türleri	20
2.4 YBM ve İşbirliği	24
2.5 YBM ve Veri	29
2.5.1 Veri, enformasyon, bilgi (data, information, knowledge).....	29
2.5.2 Mimarlık, mühendislik ve inşaat sektöründe veri.....	30
2.5.3 YBM verisi.....	32
2.5.3.1 YBM sisteminde veri kullanımı	34
2.5.3.2 YBM sisteminde veri yönetimi	38
3. HESAPLAMALI TASARIM	43
3.1 Mevcut YBM Sisteminde Veri ile İlgili Sınırlamalar	43
3.2 Görsel Programlamanın Tarihçesi.....	48
3.3 Görsel Programlama İlkeleri	49
3.4 Hesaplamalı Tasarımdan YBM'ye doğru.....	52
3.5 Bölüm Değerlendirmesi	53
4. DYNAMO İLE GÖRSEL PROGRAMLAMA	55
4.1 Dynamo'ya Genel Bakış	57
4.2 Dynamo ile Yapılan Uygulamalar.....	64
4.2.1 Analiz ve optimizasyon uygulamaları.....	65
4.2.1.1 Dynamo'nun performans odaklı tasarım için kullanılması.....	66
4.2.1.2 Dynamo'nun akustik optimizasyon için kullanılması.....	78
4.2.1.3 Optimo ve NSGA-II optimizasyonları	81

4.2.1.4 Project Fractal ile yapılan uygulamalar.....	88
4.2.1.5 Mekân planlama ve optimizasyonu (Space Planning)	96
4.2.2 Model yönetimi ve kalite kontrol uygulamaları.....	102
4.2.2.1 Revit'te yapılması gereken standardize işlerin Dynamo ile yönetilmesi	105
Bu bölümde, Revit ortamında yapılabilen fakat süreç içerisinde zaman ve emek kaybına sebep olan 12 tane iş akışının Dynamo ile nasıl daha verimli bir hale getirilebileceği uygulamalar üzerinden anlatılmaktadır.....	105
4.2.2.2 Yapı bilgi modelindeki verilerin, Dynamo ile yönetilmesi ve doğruluğunun sağlanması.....	124
4.2.2.3 Dynamo ile modeldeki verilerin görselleştirilmesi	129
4.2.3 Harici yazılım ve donanım entegrasyonu.....	132
4.3 Bölüm Değerlendirmesi.....	135
5. SONUÇ VE ÖNERİLER.....	139
KAYNAKLAR.....	145
ÖZGEÇMİŞ.....	151



KISALTMALAR

CAD	: Bilgisayar Destekli Tasarım (İng.: Computer Aided Design)
LOD	: Level Of Detail
LOI	: Level Of Information
YBM	: Yapı Bilgi Modelleme (İng.: BIM: Building Information Modeling)
MMİ	: Mimarlık, Mühendislik ve İnşaat Sektörü (İng.: AEC: Architecture, Engineering and Construction Industry)
2D	: 2 Boyutlu
3D	: 3 Boyutlu
4D	: 4 Boyutlu
5D	: 5 Boyutlu
6D	: 6 Boyutlu
7D	: 7 Boyutlu



ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1: Dynamo Paketlerinden Bazıları.....	63
--	----





ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Nesnelerin LOD seviyeleri (McPhee, 2013)	10
Şekil 2.2 : Nesnelerin LOD seviyesinin şartnamede belirtilmesi (Url-1).....	10
Şekil 2.3 : YBM'nin boyutları (Url-2).....	11
Şekil 2.4 : YBM olgunluk seviyeleri (Bew ve Richards, 2008)	13
Şekil 2.5 : YBM olgunluk seviyeleri (Url-3).....	13
Şekil 2.6 : YBM Standartları (Holzer, 2016).....	18
Şekil 2.7 : Autodesk Classification Manager eklentisi.....	23
Şekil 2.8 : YBM işbirliği bileşenleri (Koppula, 2012)	24
Şekil 2.9 : Mimari, statik ve mekanik model (Url-4)	25
Şekil 2.10 : Navisworks programı ile yapılan çakışma testleri (Url-4)	26
Şekil 2.11 : YBM ve geleneksel işbirliği sürecinin karşılaştırılması (Holzer,2008). 27	
Şekil 2.12 : BCF Manager eklentisi (Url-6)	28
Şekil 2.13 : Bilgi Hiyerarşisi (Bender ve Fish, 2000).....	29
Şekil 3.1 : Proje tasarım optimizasyonu (Aish ve diğ., 2012)	46
Şekil 3.2 : DesignScript NodeToCode özelliği ile algoritmanın dönüştürülmesi	51
Şekil 4.1 : Dynamo'nun diğer uygulamalarla olan ilişkisi. Kaynak referans alınarak yeniden yorumlanmıştır. (Sgambelluri, 2015).....	55
Şekil 4.2 : Dynamo ile problem çözme iş akışı	56
Şekil 4.3 : Dynamo ara yüzü.....	57
Şekil 4.4 : Dynamo'da oluşturulan algoritma ve geometrinin ön izlemesi	58
Şekil 4.5 : Dynamo grafiğinin düzenlenmesi (Dynamo Primer, t.y.)	59
Şekil 4.6 : Dynamo grafiği uygulama planı (Ondejcik, 2016)	60
Şekil 4.7 : Dynamo'da kod bloğunun kullanılması	60
Şekil 4.8 : Dynamo paket yöneticisi (Url-7).....	61
Şekil 4.9 : Dynamo paketlerinin indirilmesi ve kullanılması	62
Şekil 4.10 : MacLeamy eğrisi ile proje aşamalarında çaba, zaman ve maliyetin karşılaştırması (Url-8).....	65
Şekil 4.11 : Geleneksel performans odaklı tasarım iş akış şeması	68
Şekil 4.12 : Bütünleşik performans odaklı tasarım iş akış şeması.....	68
Şekil 4.13 : Beybi Giz Plaza Revit modeli	70
Şekil 4.14 : Mevcut giydirmce cephe özelliklerinin Revit'e işlenmesi.....	70
Şekil 4.15 : Insight 360 programında yapılan enerji analizi çıktıları	71
Şekil 4.16 : Aydınlatma analizi çıktıları	71
Şekil 4.17 : Insight Beybi Giz Plaza ısıtma yükleri analizi	72
Şekil 4.18 : Beybi Giz Plaza soğutma yükleri analizi	72
Şekil 4.19 : Güneşe bağlı olarak boyutları değişebilen gölge elemanı.....	73
Şekil 4.20 : Yüzeyin oluşturulması ve noktaların tanımlanması	73
Şekil 4.21 : Yüzeyin panelizasyonu.....	74
Şekil 4.22 : Cephe elemanında tanımlanan açıklık oranı (aperture ratio) parametresi.....	74
Şekil 4.23 : Gölge elemanlarının yüzeye yerleştirilmesi ve açıklık oranı parametresinin tanımlanması	74

Şekil 4.24 : Panel normallerinin ve güneş ışınlarının (vektörlerinin) arasındaki mutlak farkın bulunması	75
Şekil 4.25 : Gölge elemanın açıklık oranını belli değerler arasında tutmak için yapılan yeniden eşleme işlemi	75
Şekil 4.26 : Dynamo’da oluşturulan algoritmanın tamamı.....	76
Şekil 4.27 : Dynamo ile oluşturulan ikincil cephenin Revit ortamında görüntülenmesi.....	76
Şekil 4.28 : İkincil cephedeki gölgeleme elemanlarının açıklık oranlarının, güneşe göre farklılaşması.....	77
Şekil 4.29 : Acustamo paketine ait düğümler	78
Şekil 4.30 : Ses kaynağının temsil edilmesi (Vannini, 2015).....	79
Şekil 4.31 : Ses ışınlarının yüzeyde temsil edilmesi (Vannini, 2015)	79
Şekil 4.32 : Ses ışınlarının yüzeyde temsil edilmesi (Vannini, 2015)	80
Şekil 4.33 : Dynamo Acustamo paketi ile ses ışınlarının ve yüzeylerinin oluşturulması (Hudson ve Strathaus, 2015).....	80
Şekil 4.34 : Dynamo Optimo paketi ile tasarım alternatiflerinin denenmesi (Hudson ve Strathaus, 2015).....	81
Şekil 4.35 : Sonuç tasarım (Hudson ve Strathaus, 2015).....	81
Şekil 4.36 : Dynamo Optimo paketi düğümleri.....	82
Şekil 4.37 : YBM tabanlı performans optimizasyonu uygulama süreci (Asl ve diğ., 2015b)	83
Şekil 4.38 : Optimo çalışma yapısı (Asl ve diğ., 2015a)	84
Şekil 4.39 : YBM tabanlı performans optimizasyon çerçevesine genel bakış. (Asl ve diğ., 2015b)	86
Şekil 4.40 : Optimizasyon süreci iş akışı (Asl ve diğ., 2014).....	86
Şekil 4.41 : Tasarım alternatiflerine ait sayısal değerlerin ve üç boyutlu modellerin görüntülenmesi (Asl ve diğ., 2014).....	87
Şekil 4.42 : Project Fractal ve Dynamo Studio’nun birlikte kullanılması	89
Şekil 4.43 : Autodesk Dynamo Studio ile oluşturulan stadyum algoritması ve ön izlemesi (Autodesk Project Fractal, t.y.).....	91
Şekil 4.44 : Project Fractal değişkenler grafiği (Autodesk Project Fractal, t.y.)	91
Şekil 4.45 : Project Fractal ortamında oluşturulmuş olan tasarım alternatifleri (Autodesk Project Fractal, t.y)	91
Şekil 4.46 : Project Fractal değişkenlerinin filtrelenmesi (Autodesk Project Fractal, t.y.)	92
Şekil 4.47 : Project Fractal ortamından tasarımı dışarıya aktarma seçenekleri (Autodesk Project Fractal, t.y.)	92
Şekil 4.48 : Dynamo’da oluşturulan karma kullanımlı yapı örneği algoritması ve ön izlemesi	93
Şekil 4.49 : Dynamo’da oluşturulan algoritmaya ait girdi ve çıktı parametreleri	93
Şekil 4.50 : Project Fractal ile oluşturulan tasarım alternatifleri	94
Şekil 4.51 : Formit ortamında Dynamo parametrelerinin değiştirilmesi	95
Şekil 4.52 : Tasarım alternatiflerine ait verilerin Power BI ortamında görselleştirilmesi	96
Şekil 4.53 : Dynamo Space Planning paketi.....	96
Şekil 4.54 : Mahal, departman ilişkisi ve K-D Veri Strüktürü (Das ve diğ., 2016)...	97
Şekil 4.55 : Hiyerarşik mekân planlama yaklaşımı (Das ve diğ., 2016).....	98
Şekil 4.56 : Mekân Planlama Türeticisi için oluşturulan CSV tablosu (Das ve diğ., 2016)	98
Şekil 4.57 : Proje sınırlarının ve alt bölümlenmelerin oluşturulması (Das ve diğ., 2016)	99

Şekil 4.58 : Mekân planlama türeticisi hücre çalışma mantığı (Das ve diğ., 2016) ..	99
Şekil 4.59 : Mekân planlama türeticisi tasarım süreci (Das ve diğ., 2016)	100
Şekil 4.60 : Mekân Planlama Türeticisi ile oluşturulan plan alternatifleri (Das ve diğ., 2016)	101
Şekil 4.61 : Kat yükseklikleri bilgisinin Excel’den alınması	105
Şekil 4.62 : Oluşturulan kat seviyelerinin (level) Revit’te görüntülenmesi	105
Şekil 4.63 : Kat seviyelerinin oluşturulması için Dynamo’da hazırlanan algoritma	106
Şekil 4.64 : Kat planlarına ait verilerin Excel’den okunması.....	106
Şekil 4.65 : Kat planlarının oluşturulması	107
Şekil 4.66 : Kat seviyelerinin ve planlarının oluşturulması için yazılan algoritma.	107
Şekil 4.67 : Kat planlarının yeniden isimlendirilmesi algoritması	108
Şekil 4.68 : Tavan kat planlarının oluşturulması ve isimlendirilmesi için yazılan algoritma	108
Şekil 4.69 : Özel düğümlerin Dynamo kütüphanesine yerleştirilmesi	109
Şekil 4.70 : Özel düğümlerin düzenlenebilmesi	109
Şekil 4.71 : Revit’te scope box oluşturulması ve algoritmaya tanıtılması	109
Şekil 4.72 : Görünüşlerin oluşturulması algoritması	110
Şekil 4.73 : Pafta verilerinin Excel’den okunması	110
Şekil 4.74 : Revit ortamından antetin seçilmesi ve pafta oluşturmak için kullanılan düğüm	111
Şekil 4.75 : Revit ortamında bütün paftaların oluşturulması.....	111
Şekil 4.76 : Pafta oluşturulması için hazırlanan algoritma	112
Şekil 4.77 : Dynamo Player ile hazırlanan algoritmanın yürütülmesi.....	113
Şekil 4.78 : Dynamo Player ile girdilerin, Revit ortamında değiştirilebilmesi	113
Şekil 4.79 : Dynamo’da hazırlanan kat planlarının oluşturulması algoritması	114
Şekil 4.80 : Dynamo’da hazırlanan kat planlarının Revit’te görüntülenmesi	114
Şekil 4.81 : Oda numaralarının nasıl revize edileceğine dair hazırlanan algoritmanın çalışma mantığı	115
Şekil 4.82 : Oda parametrelerinin Revit modelinden alınması.....	115
Şekil 4.83 : Odaların yeni numaralarının oluşturulması.....	116
Şekil 4.84 : Odaların numaralarının Revit’e aktarılması	116
Şekil 4.85 : Özel düğüm oluşturma ekranı	116
Şekil 4.86 : Özel düğüm için girdi ve çıktıların tanımlanması	117
Şekil 4.87 : Oluşturulan özel düğüm ile algoritmanın hazırlanması	117
Şekil 4.88 : Revit’te oda numaraları sıralı olmayan kat planı	117
Şekil 4.89 : Revit programında eğrinin çizilmesi	118
Şekil 4.90 : Dynamo ile eğrinin seçilmesi	118
Şekil 4.91 : Dynamo ile odaların seçilmesi ve görselleştirilmesi.....	118
Şekil 4.92 : Dynamo’da eğri ve odaların kesştirilmesi.....	119
Şekil 4.93 : Odaların çizilen eğriye göre yeniden numaralandırılması	119
Şekil 4.94 : Odaların yeniden numaralandırılması için hazırlanan algoritma	119
Şekil 4.95 : Dynamo ile Excel’de hazırlanan workset verisinin okunması	120
Şekil 4.96 : “Workset.ByName” düğümü ile worksetlerin oluşturulması.....	120
Şekil 4.97 : Worksetlerin oluşturulması için hazırlanan algoritma	121
Şekil 4.98 : Modelical Paketi, “Wall Finishes by Room” düğümü ve açılımı	121
Şekil 4.99 : Revit programında oluşturulan odalar ve Dynamo programında hazırlanan algoritma	121
Şekil 4.100 : Duvar kaplamalarının oluşturulması	122
Şekil 4.101 : Odaların ve etiketlerinin merkeze getirilmesi işlemi	122
Şekil 4.102 : Dynamo’da hazırlanan algoritma	123
Şekil 4.103 : Dynamo’da hazırlanan grid oluşturulması algoritması	123

Şekil 4.104 : Dynamo ile birlikte gridlerin oluşturulması	123
Şekil 4.105 : Revit programında yer alan gereksiz odalar	124
Şekil 4.106 : Dynamo’da hazırlanan gereksiz odaların silinmesi algoritması.....	125
Şekil 4.107 : Revit programında yer alan gruplar.....	126
Şekil 4.108 : Dynamo’da grupların tekil nesnelere dönüştürülmesi için hazırlanan algoritma	126
Şekil 4.109 : Dynamo ile hazırlanan algoritma	127
Şekil 4.110 : Dynamo ile Excel’e yazılan veriler	127
Şekil 4.111 : Dynamo ile hazırlanan nesnelerin hangi worksette olduğunun raporlanması algoritması.....	128
Şekil 4.112 : Dynamo ile odaların görselleştirilmesi algoritması.....	129
Şekil 4.113 : Odaların görselleştirilmesi.....	129
Şekil 4.114 : Dynamo ile hazırlanan algoritma ve görselleştirilen cephe.....	130
Şekil 4.115 : Revit programında, uyarlanabilir bileşenlerin görüntülenmesi	130
Şekil 4.116 : Mandrill paketi ve veri görselleştirme grafikleri (Mandrill Primer, t.y.)	131
Şekil 4.117 : Mandrill paketi ile oda bilgilerinin görselleştirilmesi	131
Şekil 4.118 : Excel’den Revit programına veri alınması	132
Şekil 4.119 : Revit programından Excel’e veri yazılması	133
Şekil 4.120 : Revit ile çizelgelerin yazı formatında ihraç edilmesi	133
Şekil 4.121 : Çizelgelerin Excel’e aktarılması ve gerekli düzenlemelerin yapılması	134
Şekil 4.122 : Modelical “Schedules Exporter” düğümü	134
Şekil 4.123 : Çizelgelerin ihraç edilmesi için Dynamo’da hazırlanan algoritma	134
Şekil 4.124 : Çizelgelerin Dynamo ile birlikte Excel formatında ihraç edilmesi	135

YAPI BİLGİ MODELLEME TABANLI HESAPLAMALI TASARIM YAKLAŞIMI İLE MİMARİ TASARIM İŞ AKIŞLARININ VERİMLİLİĞİNİN SAĞLANMASI

ÖZET

Geçmişten günümüze teknolojinin gelişmesi ile birlikte mimari tasarım yöntemleri ve süreçleri farklılaşmaktadır. CAD sistemlerinin kullanıldığı dönemden Yapı Bilgi Modelleme ve hesaplamalı tasarım yaklaşımlarının kullanıldığı döneme geçiş süresince tasarımcıların mimari tasarım sürecine olan yaklaşımları değişmektedir. Geleneksel tasarım yöntemlerinde tasarımcının bilgisayarı sadece temsil aracı olarak kullanmasının ötesinde yaşanan paradigma değişimleriyle birlikte tasarımcı bilgisayarları tasarım ortağı olarak kullanarak hesaplama, analiz ve optimizasyon imkanlarından yararlanmaktadır. Mimari tasarım sürecinde tasarım problemi algoritma düzeyinde soyutlanarak sürece aktarılmaktadır. Böylece sayısal tasarım araçları ile birlikte tasarım süreci sadece sonuç ürünün ortaya koyulmasından ziyade sürecin tasarlanmasına imkân vermektedir.

Tez kapsamında öncelikle YBM sistemi, mevcut YBM yazılımlarının mimari tasarım kısıtlamaları, mimarlık, mühendislik ve inşaat sektöründe YBM tabanlı hesaplamalı tasarım yaklaşımına neden ihtiyaç duyulduğu incelenmektedir. YBM sisteminin CAD sistemlerine göre daha verimli ve bütünlük bir proje süreci oluşturduğunu açık bir şekilde söylemek mümkündür. Nesne tabanlı çalışan YBM yazılımlarının tek bir model üzerinden 2D ve 3D temsillerin elde edilebilmesi, farklı disiplinlerin ortak ve tutarlı veriler üzerinden koordine olabilmesi ve mimari tasarım süreci içerisinde elde edilen verilerin tesis yönetimine kadar modelde bulunabilmesi ve istenildiği zaman alınabilmesi gibi özellikler ile daha verimli bir proje süreci elde edilebilmektedir. YBM sistemlerinin mimarlık, mühendislik ve inşaat sektörüne getirdiği birçok avantajının olmasının yanı sıra mimari tasarım sürecinde iş gücü ve zaman harcanmasını gerektiren iş akışları süreç içerisinde hala mevcuttur. Bu durumun YBM tabanlı proje süreçlerine sahip ofisler için optimize edilmesi gereken bir süreç olarak ele alınması gerekmektedir. Bu amaçla, YBM tabanlı hesaplamalı tasarım yaklaşımı ile mimari tasarım süreçlerinde optimize edilmesi gereken iş adımlarının yeniden ele alınarak süreç içerisinde görülen zaman ve iş gücü kaybının önüne geçilerek verimliliğin sağlanabileceği düşünülmektedir.

Tez çalışmasının devamında hesaplamalı tasarım yaklaşımı olan görsel programlama ve YBM tabanlı, görsel programlama aracı olan Dynamo detaylı bir şekilde ele alınarak MMİ sektörüne sağladığı olanaklar ortaya konulmaktadır. Mevcut YBM yazılımı olan Revit ve Dynamo'nun birlikte çalışması ile optimize edilebilecek olan iş akışları ve mimari tasarım süreci için sağladığı olanaklar uygulamalar üzerinden aktarılmaktadır. Projenin erken tasarım aşamasından tesis yönetimine kadar olan süreçte piyasada görülen problemlere Dynamo'nun getirdiği farklı çözüm yaklaşımları değerlendirilmektedir. Dynamo ile birlikte geometrik temsil, mekân planlama, parametrik tasarım, performans ve strüktürel analizler, bina ve model optimizasyonu ve veri paylaşımı gibi alanlarda araştırmalar yapılarak Yapı Bilgi Modelleme ortamına

nasıl katkıda bulunduđu incelenmektedir. Dynamo ile yapılan uygulamalar, *analiz ve optimizasyon uygulamaları, veri yönetimi ve kalite kontrol ile veri alışverişı için harici yazılımlarla oluşturulan entegrasyon* olmak üzere üç ana başlık altında detaylandırılmaktadır. Bu başlıklar detaylandırılırken yapılan uygulamalarda Dynamo'dan başka eklentiler de kullanılmaktadır. Çünkü Dynamo, paket yöneticisine kullanıcılar tarafından eklenen yeni araçlar ve Autodesk bünyesinde geliştirilen bulut sistemleri gibi yan ortamlarla daha donanımlı hale gelmektedir. Dynamo'nun açık kaynak kodlu olması diğer araçlarla birlikte gelişmesinin önünü açmaktadır. Dynamo'nun gelişmesine katkı sağlayan yan ortamlar ele alınan örneklerle birlikte ilgili bölümlerde anlatılmaktadır.

Sonuç olarak, hesaplamalı tasarım yaklaşımlarının YBM tabanlı mimari tasarım süreçlerine sağladığı katkılar değerlendirilerek mimari tasarım iş akışlarının verimliliğinin sağlanabilmesi için bir yaklaşım önerisi sunulmaktadır. Bu tez çalışması ile ofislerin YBM tabanlı mimari tasarım süreçlerini optimize edebilmeleri ve bu alanda yapılacak olan gelecekteki çalışmalar için zemin oluşturması amaçlanmaktadır.



ENSURING THE EFFICIENCY OF ARCHITECTURAL DESIGN WORKFLOWS WITH COMPUTATIONAL DESIGN APPROACH BASED ON BUILDING INFORMATION MODELING

SUMMARY

With the development of technology from the past, the methods and processes of architectural design are different. During the transition period in which CAD systems are used, the use of the BIM (Building Information Modeling) and computational design approaches is changing the way designers approach the architectural design process. Beyond using the computer as a means of representation in the traditional design methods, the designer takes advantage of the computing, analysis and optimization opportunities by using the designer computers as design partners with the paradigm changes. In the architectural design process, the design problem is abstracted at the level of the algorithm and transferred to the process. Thus, together with the digital design tools, the design process allows for the design of a continuous process, not just the resultant product.

Architectural design process is in deepening relations with developing technology systems. In the past, designers used imagination and lines as the only means to create and represent their projects. This led many architectural design decisions to be based on the designer's intuition and experience. However, designs that can be visualized in the mind of the designer had to be shared and understood with different disciplines in the process. This situation reveals the representative issue of designs.

When the building industry begins to digitize, the traditional drawing table has become a two-dimensional computer aided design (CAD) system. With the spread of computers and the emergence of CAD software, the use of digital tools in the design process has begun. Then, in addition to the 2D CAD system, 3D features are added.

Today, computers are evolving and the architecture, engineering and construction sectors are experiencing a paradigm shift as the transformation of information age. Now, computers and software can handle large amounts of data. However, CAD systems have begun to take the place of CAD systems in the building industry. Thus, the building industry has faced a new transformation, resulting in new workflows and roles. This transformation in the architecture, engineering and construction sector has rapidly spread to the international arena, and in some countries it has become compulsory to use the BIM system, especially in public projects.

In parallel with the development of the BIM system and the development process, the computational design approach has also found a wide use in the sector. While computational design and BIM systems continue to evolve in different environments, they have been brought together on a common platform in line with the needs of the sector.

The most important feature of the BIM system is that it provides a data-rich environment for users. The BIM system is a must-have model for adding data, processed, edited in the system, and can be retrieved by the necessary disciplines

whenever required. Existing BIM software provides limited capabilities such as manipulating, manipulating and optimizing data in the system. The relationship between BIM and computational design is very interesting with the increase of data load in BIM models, the difficulty of easy adaptation of model against changes in design decisions, the difficulty of getting desired data from stratified database, the increase of number of error that user is insufficient to manage this process. In response, Dynamo, the BIM-based computational design tool, emerged. Dynamo and Grasshopper can show a strong resemblance. Both computational design tools allow the algorithms to be generated and manipulated using visual programming language. While Grasshopper is often used to manage geometric relationships that affect design morphology, Dynamo provides users with the ability to manipulate project data. So Dynamo makes much more copies of Grasshopper than copying it.

The Dynamo tool is increasingly in use as the BIM system allows it to create its own database and manage this data for specific purposes. Dynamo provides many possibilities for processing, managing and interacting with BIM data and interacting with external sources. This allows a strong association between modeling and data management processes.

While the existing methods used in architectural, engineering and construction sectors in BIM-based architectural design processes seem to cause time and cost loss, the need for productivity of workflows has arisen. Based on this requirement, the thesis explores what workflows are required to achieve productivity and time loss in BIM-based architectural design processes. In addition, the analysis and optimization that can be done in order to make design decisions reliably in the process of architectural design and to build sustainable constructions can be carried out together with Dynamo. In this context, the thesis study is carried out by integrating the computational design approach with the BIM architectural design process, and with the new workflows obtained, the process efficiency can be ensured.

Within the scope of the thesis, firstly the problems of the BIM system, constraints of project processes of existing BIM software, why BIM-based computational design approach is needed in architecture, engineering and construction industry are examined. It can be clearly stated that the BIM system creates a more efficient and integrated project process than the CAD system. A more efficient project process can be achieved with features such as 2D and 3D representations can be obtained from a single model of the object-based BIM software, coordination of different disciplines over common, consistent data and the data obtained within the project process can be found in the model up to the facility management. In addition to the many advantages that the BIM systems bring to the architectural, engineering, and construction sectors, workflows that require labor and time are still available in the process. This should be considered as a process that must be optimized for offices with BIM-based project processes. For this purpose, it is thought that by using the computational design approach based on the BIM, the work steps to be optimized in the project processes can be reconsidered and the loss of time and labor seen in the process can be prevented.

In the continuation of the thesis study, Dynamo, a visual programming tool is provided with in detail and offers possibilities for the AEC Industry. The possibilities for workflows and architectural design processes that can be optimized with the collaboration of Revit and Dynamo are transferred through applications. From the early design stage of the project to the facility management, the different solution approaches brought by Dynamo software are evaluated in the problems seen in the industry. With Dynamo, it is investigated how it contributes to the Building Information Modeling environment such as geometric representation, space planning,

parametric design, performance and structural analysis, building and model optimization and data sharing. Applications made with Dynamo are detailed under three main headings: *analysis and optimization applications*, *data management and quality control*, and *integration with external software for data exchange*. While samples are being made, other software (add-on, plugins etc.) are used except Dynamo software. Because Dynamo is becoming more equipped with side media such as new tools added to the package manager by users and cloud systems developed within Autodesk's own. The fact that Dynamo is open source is the way forward for its development. The additions that contribute to the development of Dynamo software are described in the relevant chapters with the examples discussed.

It is possible to create new workflows related to how the effort and time spent during the work carried out with the existing BIM can be minimized together with the calculated design tool Dynamo. Within the scope of the thesis, the difficulties faced by users in the AEC sector during the project process from the early design stage to the delivery phase of the architectural design process are revealed by the knowledge gained from literature research and office experience. In the early design phase, users can access many design alternatives with productive design methods by integrating Dynamo into existing BIM systems. In the early design phase, the user can make data-oriented design decisions with analysis and optimizations made in Dynamo. It is aimed to reduce the increased project costs due to the changes in the decision while reducing the number of decision changes made in the early design stage according to analyzes and optimizations and progressing phases of the project. It is difficult to manage the model with factors such as the size of the project from the early design stage to the application stage, the amount of data increased with the model's maturity level, the detail level of the project, and the number of stakeholders. With project revisions, the number of repetitive tasks is increasing. The number of errors that are made depending on the user during the revision work is also increasing. As a result, with Dynamo, a computational design tool, it is aimed to reduce the time loss and workload caused by the increasing data in the model during the application phase of the project. Moreover, with Dynamo, data loss in the project processes can be prevented by managing the data in the model. As a result, features such as Dynamo's visual programming language, BIM-based architecture, engineering, and construction can be supplemented with tools for optimizing workflows in offices. With Dynamo, users can perform tasks related to data management, data manipulation, data transfer and documentation work in the structure information model of the project during application phase while performing more analysis and optimization in conceptual design phase.

In conclusion, a proposal is proposed to ensure the efficiency of architectural design workflows by evaluating the contribution of computational design approaches to BIM-based project processes. It has been researched how Dynamo current BIM-based architectural design workflows can be achieved within the scope of the thesis. The literature survey in this context shows that the researcher can avoid the time and unnecessary loss of work power in the current BIM processes with the computational design approaches along with his own office experience and applied examples. In addition, the success of the accurate modeling that the data management brings together with the computational design approach reflects every phase of the project process. With this master thesis study, it is aimed that the offices can optimize BIM-based architectural design processes and a source for future work on this area.



1. GİRİŞ

1.1 Tezin Amacı ve Kapsamı

Yapı Bilgi Modelleme'nin (YBM) ortaya çıkışı, mimarlık, mühendislik ve inşaat sektöründe artan kullanımı YBM'de giderek artan miktarda oluşan verinin nasıl ele alınacağı ve kullanılacağı üzerine yeni arayışlara sebep olmuştur. YBM etkin araçları ile tüm veriler, bir projenin dijital bir kopyasını temsil eden bir veri tabanı oluşturarak birbirleriyle doğrudan veya dolaylı bir ilişkiye sahiptir. Veri tabanı, dijital bir model olarak temsil edilmekte ve modelde yer alan YBM nesnelere yerleştirilmiş birçok nitelik barındırmaktadır. YBM'nin mimarlık, mühendislik ve inşaat sektörüne getirdiği birçok kolaylık ve yenilik olmasına rağmen YBM araçlarıyla yürütülen mimari tasarım süreci içerisinde el emeği gerektiren birçok iş adımı bulunmaktadır. Verilerin modele eklenmesi, yönetimi ve istenilen verilerin modelden çıkartılması sistem içerisinde manuel bir şekilde yapılmaktadır. Bu durum kullanıcıların standardize işler için zaman ve emek harcamasına sebep olurken asıl yoğunlaşılması gereken tasarım problemlerine harcanması gereken zamanı azaltmaktadır.

YBM nesnelerinde belirli bir dereceye kadar parametrik zekâ ve bağımlılıklar bulunmaktadır. Model genelinde nesnelerin birbirleriyle olan ilişkileri anlamında parametrik ilişkiler kurgulanamamaktadır. Ancak algoritmaların kullanılması ile tüm modele yayılan zekâ ve parametrik ilişkiler mevcut YBM iş akışına entegre edilebilmiştir. Müşterilerin yüksek beklentileri, tasarım alanlarının ve koşullarının karmaşıklığı ile mimari tasarım iş akışlarının verimliliğinin sağlanması ihtiyacı yeni veri yönetimi ve veri düzenleme araçlarına olan ihtiyacı doğurmuştur. Bu ihtiyaca yönelik olarak görsel programlama dilleri kullanılmaya başlanmıştır. Görsel programlama, metin tabanlı programlama dillerinin esnekliğini kullanarak, belirli işlevleri birbirlerine görsel olarak bağlayan grafik düğümler ile algoritmaların oluşturulmasıdır. Böylece, programcı olmayan kullanıcılar, soyut sözdizimleri ile karşı karşıya kalmamakta ve yapılan eylemlerin eş zamanlı olarak görsel geribildirimini alınabilmektedir.

Görsel programlama ve Yapı Bilgi Modelleme, birbirinden ayrı ortamlar olarak gelişirken, YBM ortamında oluşan tasarım süreçlerinin ve eylemlerinin analiz edilebilmesi, otomatikleştirilebilmesi, veri yönetiminin sağlanarak süreç içerisindeki veri kayıplarının en aza indirgenebilmesi için ortak bir platformda buluşturulmuştur. YBM tabanlı hesaplamalı tasarım ile YBM süreçlerindeki iş akışları dönüştürülmüştür. YBM tabanlı hesaplamalı tasarım aracı olarak geliştirilen Dynamo ile YBM araçlarında görülen mevcut sınırlamaların aşılması ile birlikte YBM potansiyelinden daha fazla yararlanılması istenmektedir. Dynamo'nun mimarlık, mühendislik ve inşaat sektöründe görülen birçok probleme cevap olarak kullanılması ve açık kaynak kodlu olması sebebiyle Dynamo için birçok özel paket geliştirilmiş ve geliştirilmeye devam edilmektedir. Ayrıca bulut tabanlı sistemler gibi birlikte çalışabileceği harici yazılımlar ile birlikte desteklenmektedir. Böylece, YBM ve hesaplamalı tasarımın entegrasyonunun geliştirilmesi ve sektörde görülen yaygın problemlerin optimize edilebilmesi için birçok farklı yazılımın beraberinde getirdiği yeni iş akışları oluşturulmuştur. Tez kapsamında, hesaplamalı tasarımın YBM sürecine entegre olması ile birlikte ortaya çıkan yeni iş adımlarının neler olduğu, mevcut YBM sürecinde görülen kısıtlamaların hesaplamalı tasarım ile birlikte nasıl aşılabileceği araştırılmaktadır. Bu kapsamda literatür araştırması ve araştırmacının kendi çalışma deneyimlerinden yola çıkılarak YBM sürecinde kullanıcının yoğun zaman ve iş gücü harcadığı işlerin, hesaplamalı tasarım ile nasıl otomatikleştirilebileceği konusu üzerine bir yöntem araştırması hedeflenmektedir. Tez kapsamında YBM sürecinin temelini oluşturan “veri” üzerine ayrıca araştırma yapılmaktadır. Bunun sebebi, YBM sürecinde veri oluşturmanın, yönetiminin zorunlu olması ve kullanıcıların verilerin yönetimi ile ilgili yapılması gereken işlere oldukça fazla zaman ve emek ayırmak zorunda kalmalarıdır.

Ayrıca, tez kapsamında Dynamo'nun mevcut YBM sürecine getirdiği yeni olanakların araştırılması konusunda Dynamo ile ilgili ortaya konulmuş akademik çalışmaların sınırlı olması da etkili olmuştur. Dynamo ile ilgili topluluk tarafından oluşturulmuş internet kaynaklı araştırmalar ile Autodesk University tarafından yayınlanan eğitimler yoğunluktadır. Bu nedenle tezde ele alınan Dynamo'nun YBM iş akışlarının geliştirilerek mimari tasarım sürecinin verimliliğinin sağlanması konusunun kapsamlı bir şekilde araştırılmasının, YBM sistemini kullanan ofisler ve kullanıcılar için faydalı olacağı düşünülmektedir.

Tez kapsamında ařağıdaki sorulara cevap aranmaktadır:

1. Mevcut YBM sisteminde görölen sınırlamalar nelerdir?
2. Hesaplama tasarım yöntemi olan görsel programlama ile YBM'nin iş akışı iyileştirilebilir mi?
3. Görsel programlama hangi amaçlar için YBM sistemine entegre olabilmektedir?
4. Görsel programlamanın kullanılmasının önündeki engeller nelerdir? Bu engeller nasıl aşılabilmektedir?
5. Görsel programlamanın YBM ile entegrasyonu, hangi görevler için kullanılmaktadır?
6. Görsel programlama kullanımının şirket içinde kolaylaştırılması ve anlaşılabilmesi için neler yapılabilmektedir?

1.2 Tezin Yöntemi

Tez kapsamında ilk olarak literatür araştırması yapılmıştır. “Yapı Bilgi Modelleme” kavramı, sınırlamaları, hesaplamalı tasarım yöntemi olan görsel programlama ve YBM sürecine entegre olması ile birlikte gelişen iş akışları araştırılmıştır. Literatür araştırmasından ve araştırmacının kendi deneyimlerinden edindiğı bilgiler, tez kapsamında yapılan uygulamalar ile test edilmiştir. Tez boyunca ele alınan konuların uygulamasının yapılması ile Dynamo görsel programlama aracının YBM ortamı olan Revit programı ile kurduğı ilişkinin iş akışlarını nasıl değıştirdiğinin avantajları ve dezavantajları ortaya koyularak bir tartışma ortamı amaçlanmıştır. Yapılan uygulamalar probleme yönelik hazırlanan Revit ve Dynamo uygulamalarından oluşmaktadır.

Tez 5 bölümde ele alınmaktadır. İlk bölümde araştırmanın sebepleri, amaçları, araştırma soruları, kapsamı ve yöntemi tanıtılmaktadır. İkinci bölüm de YBM'nin tanımı yapılarak proje süreçlerindeki kullanımı değıerlendirilmektedir. Ayrıca YBM sisteminin olmazsa olmazı olan veri ile ilgili kullanımı, yönetimi ve disiplinler arasındaki paylaşımı ayrıntılı olarak incelenmektedir. Üçüncü bölümde mevcut YBM ile ilgili sınırlamaların neler olduğı, hesaplamalı tasarım yöntemlerinden görsel

programlama ve YBM tabanlı hesaplamalı tasarıma neden ihtiyaç duyulduđu ele alınmaktadır. Dördüncü bölümde YBM tabanlı hesaplamalı tasarım aracı olan Dynamo detaylı bir şekilde incelenmektedir. Bu bölümde Dynamo'nun gelişimi, kullanımı ve mevcut YBM sistemine nasıl entegre olduđu araştırılmaktadır. Araştırma kapsamında Dynamo'nun potansiyellerinin keşfedilmesi için sektörde görülen problemlerin çözümüne yönelik olarak çeşitli uygulamalar hazırlanmıştır. Yapılan uygulamalar mevcut YBM sistemlerine getirdikleri yenilik ve avantajlara, çözdükleri problem türüne ve kullanıldıkları proje aşamasına göre sınıflandırılarak yapılmaktadır. Dynamo ile yapılan uygulamalar ile yeni iş akışları ve potansiyeller değerlendirilmektedir. Beşinci bölüm olarak ele alınan değerlendirme kısmında tezde elde edilen bulguların tezin amacına yönelik olarak incelemesi yapılmaktadır.



2. YAPI BİLGİ MODELLEME (YBM)

Yapı Bilgi Modelleme, katı modelleme programlarının ortaya çıktığı 1970'li yıllara, ArchiCAD yazılımının geliştirildiği 1980'lere ve Revit yazılım programının geliştirildiği 2000'lere kadar uzanan bir geçmişe sahiptir. Entegre ve birlikte çalışabilir yazılım ihtiyacı 1970'lerde ortaya çıkmıştır (Björk, 1995). Mimarlar ve mühendisler, geleneksel çizim ve hesaplama araçlarını CAD kullanımından YBM sistemlerine doğru kaydıldılar (Czmoch ve Pękala, 2014). YBM, CAD sistemlerine göre daha fazla “zekâ” ve karşılaştırılabilir bilgi sağlamaktadır (Miettinen ve Paavola, 2014).

Yirmi birinci yüzyılın ilk yıllarında, mimarlar, mühendisler ve inşaatçılar için yeterli depolama ve grafik yetenekleri ile güçlü, ölçeklendirilmiş bilgisayarların geniş kullanılabilirliği, diyagram olarak iki boyutlu çizimlerden temsil oluşturma yönteminden üç boyutlu dinamik temsil araçlarına geçişe sebep olmuştur. Bu değişim, üretimde iyi anlaşılan dijital modelleme ve bilgisayar kontrollü imalatın daha modern paradigmalara sebep olmasından dolayı önemlidir. Bu modellere dayanan bina tasarımı, tasarım metodolojisindeki güçlü değişikliklerle, özellikle simülasyon ve analizin kolay kullanılabilirliği, optimizasyon ve yapı temsili tarafından dönüştürülmektedir. Bu teknolojiler tasarım eyleminin kendisini ve sonuçlarını yeniden tanımlamaktadır.

Eastman ve diğerleri (2011), Yapı Bilgi Modelleme’yi bir modelleme teknolojisi ve yapı modellerini üretmek, iletmek, analiz etmek için ilişkili süreçler kümesi olarak tanımlamaktadır. Yapı bilgi modelleme şu özelliklere sahiptir:

- Yazılım uygulamalarını tanımlayan hesaplanabilir grafik ve veri özelliklerini taşıyan dijital temsiller (nesneler) olarak gösterilen yapı bileşenlerinin yanı sıra akıllı bir şekilde manipüle edilmesine izin veren parametrik kurallar.
- Analiz ve iş süreçleri için gereken veri. Örneğin; metraj, teknik özellikler ve enerji analizi gibi bileşenler.
- Bileşen verisine göre değişen tutarlı verilerin, bileşenin ve bileşenin parçası olduğu montajların tüm görünümünde temsil edilmesi.

- Verilerin modelin tüm görünüşlerinde koordineli bir şekilde temsil edilmesi.

Yapı Bilgi Modelleme, AIA (2007) tarafından geliştirilen terimler sözlüğünde üç şekilde tanımlanmaktadır:

1. Birlikte çalışabilirlik için, açık standartlar üzerine kurulu, bir yapının fiziksel ve fonksiyonel özelliklerinin paylaşılan sayısal temsili.
2. Yapının ömrü boyunca, karar alınabilmesi için güvenli bir temel oluşturan, yapı hakkında paylaşılan bilgi kaynağı.
3. Farklı disiplinlerin rollerini destekleyecek şekilde bilginin eklenmesi, çıkartılması, yenilenmesi ve değiştirilmesi amacıyla yapının ömrünün farklı aşamalarında disiplinler arasındaki işbirliğinin sağlanması için kullanılan ortak veri tabanı.

YBM, bir projede iletişimin desteklenmesi (veri paylaşımı) için proje yaşam döngüsü boyunca çok boyutlu veri açısından zengin görünüm oluşturarak ve sürdüren bir temsil sürecidir; işbirliği (paylaşılan veriler üzerinde çalışılması), simülasyonlar (tahminler için verilerin kullanılması) ve optimizasyon (tasarımı, dokümantasyonu ve teslim ürününü iyileştirmek için geri bildirimin kullanılması) süreçlerinden oluşmaktadır (Mihindu ve Arayıcı, 2008).

Vektörlerden oluşan mimari çizimlerde geleneksel iki boyutlu Bilgisayar Destekli Tasarım'ın (Computer Aided Design-CAD) aksine, üç boyutlu nesne tabanlı mimari çizim yapmak için duvarlar, sütunlar, zeminler, kapılar, pencereler ve diğer birçok yapı bileşeni gibi bileşenler kullanılmaktadır. Ayrıca, konum bilgisi, malzeme, performans ve kullanıcı bilgisi gibi kapsamlı bilgi türlerini destekleyen çeşitli özellikler de model içerisinde barındırılmaktadır (Yoon ve diğ., 2009).

YBM, yeni bir yapı tasarım ve belgeleme yöntemidir. YBM terimi, bir yapının tasarım, üretim ve yönetim bilgisinin oluşturulmasını ve kullanılmasını içermektedir. YBM yazılımları, nesne yönelimli (object-oriented) yöntemler kullanılarak geliştirilmektedir. Haagenrud ve diğerlerine (2008) göre YBM'nin içeriği şunlardan oluşmaktadır:

- Fiziksel nesneler (kapı, pencere, kolon, giriş vb.)

- Mekanlar (tüm iç ve dış mekanlar)
- Tasarım, üretim ve bakım süreçleri
- Katılımcılar (organizasyonlar, firmalar, kişiler)
- Nesneler arasındaki ilişkiler

YBM, geometri, mekânsal ilişkiler, coğrafik bilgi ve yapı elemanlarının özelliklerini hep birlikte içermektedir. Ayrıca, analiz ve modelleme sonuçlarını da saklayabilmektedir. Bunların yanı sıra, bir yapının yaşam döngüsüyle ilgili her türlü bilgi YBM’de tanımlanabilmekte ve saklanabilmektedir. YBM, tüm verilerin üç boyutlu ve dijital olarak saklandığı bir havuzdur. Tekil bir veri tabanı olarak çalışan YBM’ye ortak ve genel verilerin dışında, katılımcılar tarafından da veriler eklenebilmektedir. Ayrıca, katılımcılar bu veri tabanına istedikleri zaman ve istedikleri yerden ulaşabilmektedir (Haagenrud ve diğ., 2008).

Yapı bilgi modelleme (YBM), tüm tasarım ve inşaat ekibine gerçek yapımdan önce karmaşık yapılaşma sürecini dijital olarak koordine etme yeteneği sağlamaktadır. 1980’lerde ve 1990’larda tasarım pratiğine getirilen teknolojik ilerlemelere dayanan yeni bir tasarım metodolojisi olarak ortaya çıkan YBM, tasarımcıya, karmaşık bilgisayar grafik araçlarını kullanarak, başlangıç boyutlandırma aşamasında, projenin daha birçok yönünü incelemesine izin vermektedir. Bilgisayarda sadece teslim belgelerinin çizilmesine izin veren bilgisayar destekli çizimden farklı olarak, YBM gerçek zamanlı veri tabanlarıyla üç boyutlu geometriyi birbirine bağlamaktadır. Bu tek paylaşımlı bilgi modeli sayesinde tasarım ekibi, proje sahasında faaliyet göstermeden önce, inşaatın tüm yönlerini yenileyebilmekte ve test edebilmektedir. İnşaattan hemen önce projede yer alan sorunların düzeltilebilmesiyle birlikte mimar, genel yüklenici ve mal sahibine zaman ve paradan tasarruf ettirilmektedir. YBM, sadece binaların nasıl inşa edildiğini değil, aynı zamanda binaların nasıl tasarlandığını da etkileyen bir teknolojidir. YBM’nin ortak bir bilgi modeli olması ile yapı ve tasarım bir araya getirilirken, aynı zamanda mimarın tasarım niyetini ve projenin genel özelliklerini açıkça ortaya koyabilmesini sağlamaktadır.

Günümüzde piyasada çeşitli YBM platformları bulunmaktadır. Günümüzde mevcut olan YBM platformlarının çoğu, mimarlar, mühendisler, yükleniciler ve imalatçılar gibi hizmet sağlayıcılara yönelik olup; özellikle yapı sahiplerini hedeflememektedirler (Eastman ve diğ., 2011). Mevcut YBM araçları, farklı ve çok sayıda kullanıcıya

pazarlanan çeşitli araç işlevselliğine sahiptir. Revit, Bentley System, ArchiCAD piyasadaki YBM yazılımlarından birkaçıdır.

Revit, mimari tasarımda YBM'nin yaygın olarak kullanılan yazılımı olarak kabul görmektedir. Bu platform, şu anda Revit Architecture, Revit Structure ve Revit MEP programlarını içeren entegre ürünler ailesidir. Bir araç olarak Revit, kullanıcılara kullanımı kolay bir arayüz sağlamaktadır. Bir platform olarak Revit, en büyük ilişkili uygulamalar kümesine sahiptir; tesis yönetimi gibi (Eastman ve diğ., 2011).

Mimarlık alanında bilişim ve ilgili teknolojileri benimsemek, proje süreçlerindeki verimliliği artırmaktadır. YBM özellikle, tasarım çözümlerinin üretilmesi, iletilmesi ve analiz edilmesi sürecinde modelleme teknolojisi olarak, verimliliğin artırılması için bir yardımcı olarak görülmektedir. Ancak, proje büyüklüğüne bağlı olarak artan veri miktarı ile birlikte YBM tabanlı proje süreci yönetilmesi zor bir süreç haline gelebilmektedir. YBM sürecine verilerin doğru bir şekilde eklendiğinden emin olmak gerekmektedir. Verilerin kontrolsüz bir şekilde sistem içerisine gömülmesi süreç içerisinde kontrol edilemez bir hal alabilmektedir. YBM sürecinde veri yönetiminin dışında nesnelerin geometrilerinin süreç içerisinde ne kadar detaylı modellenmesi gerektiğinin düşünülmesi gerekmektedir. Bu nedenle YBM süreçlerini verimli ve doğru bir şekilde yönetebilmek için belirli standartlara ihtiyaç duyulmaktadır.

2.1 YBM Temsil Seviyeleri

YBM ortamında farklı proje aşamalarında modelde kullanılacak olan nesnelerin detay seviyesinin, modellemenin boyutunun ve sürecin olgunluk seviyesinin tanımlanması gerekmektedir.

2.1.1 Projenin detay seviyesi

Yapı bilgi modelinde yer alan nesnelerin, bileşenlerin grafiksel anlamda içermesi gereken detay seviyeleri, LOD (Level Of Development) ile tanımlanmaktadır. Modellerin grafiksel olmayan içeriği ile ilgili model bilgisi seviyeleri LOI (Level of Information) olarak tanımlanmaktadır. Yapı bilgi modelleri için kullanılan beş temel LOD detay seviyesi vardır (Şekil 2.1). Projenin ihtiyaçları doğrultusunda YBM detay seviyesi olarak, LOD 250 veya LOD 350 gibi farklı seviyelerin kullanılması da mümkündür. Projenin hangi detay seviyesinde modelleneceğine karar verirken projenin ihtiyaçlarına ve oluşturulacak olan veri yüküne göre karar vermek

gerekmektedir. Gerekmediği müddetçe modeli gereksiz bilgiyle doldurmamak için istenilen LOD seviyeleri dikkate alınarak modelleme yapılmalıdır (Kensek, 2014). LOD detay seviyelerinin tanımları aşağıda yer almaktadır:

LOD 100: Modelde yer alan nesneler, bir sembol veya başka genel ifade şekilleriyle temsil edilebilir. Elemanların kütle olarak modellendiği seviyedir. Model elemanlarına kısa notlar eklenebilir. Veri türünün nedeniyle sadece kavramsal analizler yapılabilmektedir.

LOD 200: Modellenen elemanlara ait miktar, boyut, konum ve yönlenme açısından tahmini bilgilerinin tanımlandığı seviyedir. Bileşen bir nesne olarak oluşturulur. Bileşenlere ait basit tahmin ve çizelgeler oluşturulabilmektedir.

LOD 300: LOD 200 detay seviyesinde modellenen nesnelere ait bilgilere göre kıyaslandığında LOD 300 detay seviyesinde bilginin doğruluğundan emin olunmaktadır. Bu detay seviyesi nesnelere ait kesin bilgilerin tanımlandığı evredir. Model bu detay seviyesinde detaylı analizlerin yapılması için kullanılabilir.

LOD 400: Modellenen elemanlara ait imalat, kurulum, üretici firma bilgileri ile model numarasının sisteme eklendiği seviyedir. İnşaat yapım aşamasında görev alacak olan paydaşlar (müteahhit vs.) tarafından LOD 400 seviyesindeki model ihtiyaçlar doğrultusunda kullanılabilir.

LOD 500: Bu detay seviyesinde modellenen nesnelerin geometrileri doğrulanmaktadır. Ayrıca nesneler, tesisin işletilmesi ve bakımı için yararlı olan nitelikleri içermektedirler. Bu evre yapının işletmeye açıldığı evredir. Bu nedenle modeldeki nesnelere ait nitelikler nihai durumu yansıtmaktadır.

Projeye ait detay seviyesi genellikle proje ilerledikçe artmaktadır, yapı bilgi modeli proje başında az miktarda veri içerirken proje sürecinde sanal inşaat modeline doğru dönüşürken daha sonra varlık bilgi modeline dönüşmektedir. Modelin farklı yönleri farklı oranlarda gelişebilir, bu durum işverenin, farklı paydaşları nasıl yönlendirdiği ile yakından ilgilidir. Bu nedenle işverenin, modelin geliştirilmesinin her aşamasında gerekli olan ayrıntı düzeyini tanımlaması önemlidir. Şekil 2.2’de yer alan şartname örneğinde, sınıflandırma sistemine göre tanımlanan modeldeki elemanların hangi detay seviyesinde olması gerektiği ve buna bağlı olarak hangi bilgilerin nesnelere eklenmesi gerektiği açıklanmıştır. Bu durum sadece tasarımın yeterli detayda

gelişmesini sağlamakla kalmayıp, projenin verimli bir şekilde yürütülebilmesi için karar verilmesi sırasında gerekli olan bilginin modelde olmasını sağlamaktadır.

LEVEL of DEVELOPMENT

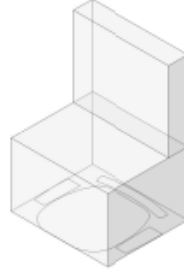
LOD 100

LOD 200

LOD 300

LOD 400

LOD 500



Concept (Presentation)

Design Development

Documentation

Construction

Facilities Management

DESCRIPTION:
Office Chair
Arms, Wheels
WIDTH:
DEPTH:
HEIGHT:
MANUFACTURER:
Herman Miller, Inc.
MODEL:
Mirra
LOD:
100

DESCRIPTION:
Office Chair
Arms, Wheels
WIDTH:
700
DEPTH:
450
HEIGHT:
1100
MANUFACTURER:
Herman Miller, Inc.
MODEL:
Mirra
LOD:
200

DESCRIPTION:
Office Chair
Arms, Wheels
WIDTH:
700
DEPTH:
450
HEIGHT:
1100
MANUFACTURER:
Herman Miller, Inc.
MODEL:
Mirra
LOD:
300

DESCRIPTION:
Office Chair
Arms, Wheels
WIDTH:
685
DEPTH:
430
HEIGHT:
1085
MANUFACTURER:
Herman Miller, Inc.
MODEL:
Mirra
LOD:
400

DESCRIPTION:
Office Chair
Arms, Wheels
WIDTH:
685
DEPTH:
430
HEIGHT:
1085
MANUFACTURER:
Herman Miller, Inc.
MODEL:
Mirra
PURCHASE DATE:
01/02/2013

(Only data in red is useable)

practicalBIM.net © 2013

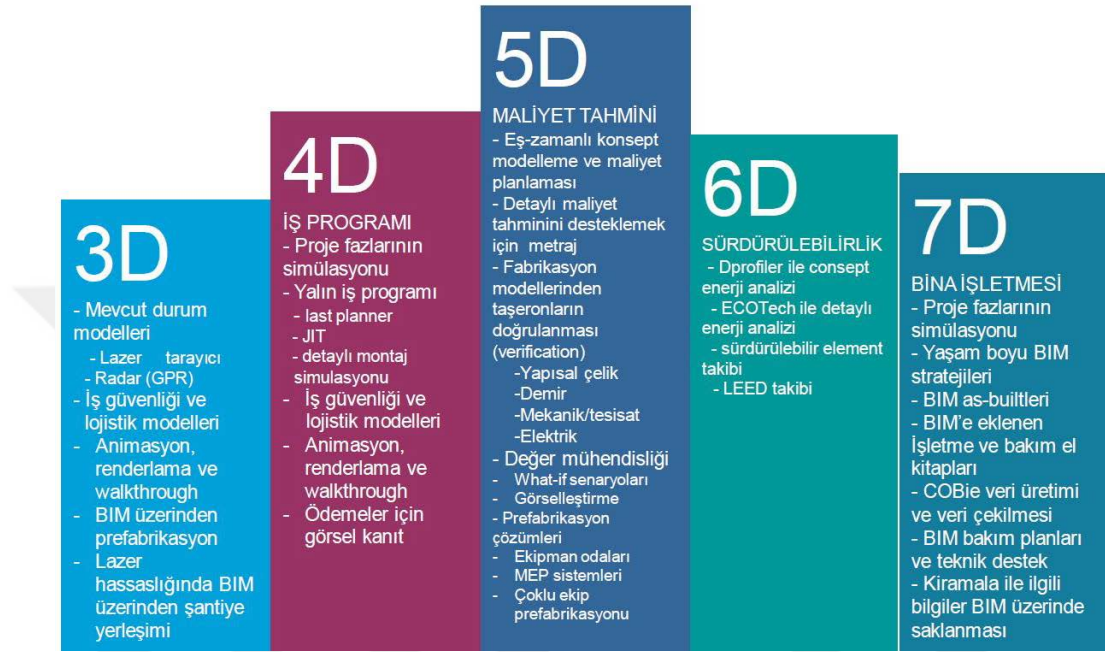
Şekil 2.1 : Nesnelerin LOD seviyeleri (McPhee, 2013)

Introduction		The Introduction has been removed and provided in a separate document – <i>LOD Spec 2017 Guide</i> . This Document also includes white papers on use of the spec.
Executive Summary		Added
Fundamental LOD Definitions – LOD 300		Call for element location added
Unifomat	Omniclass	
N/A	36-51 73 11 13 11 19	Spaces – Added
N/A	36-51 73 11 13 17 11	Horizontal Grids – Added
N/A	36-51 73 11 13 17 13	Vertical Levels - Added
A10	21-01 10	Foundations – LOD 200. Provision for location added
A1020.10.10	21-01 10 20 20	Helical Piles and Piers – Added
A4010	21-01 40 10	Standard Slabs-on-Grade – LOD 300. Provision for openings added.
B1020	21-02 10 20	Roof Construction – logic clarification added: - If modeling roof as a single composite element including structure use B1020 - If modeling individual layers as separate elements use: - B1020.10 Roof Structural Frame

Şekil 2.2 : Nesnelerin LOD seviyesinin şartnamede belirtilmesi (Url-1)

2.1.2 Modelleme boyutu

Yapı bilgi modelleme sisteminde yer alan 3D modelinin yanı sıra modele ekstra boyutlar katmak, modelde anlaşılabilirlik potansiyelini artıran faktörlerden biridir. Modele eklenen her bir boyut modelin potansiyelini artırırken yapılan analizlerin sayısını artırmaktadır. Yapı bilgi modellemesi boyutları Şekil 2.3’de gösterilmektedir.



Şekil 2.3 : YBM'nin boyutları (Url-2)

3D YBM Modeli: 3D modeli en sık rastlanan yapı bilgi modelleme boyutudur. Grafiksel ve grafiksel olmayan bilginin oluşturulması ve bu bilginin ortak bir model üzerinden paylaşılması 3D boyutuna sahip yapı bilgi modeli ile mümkündür. Proje yaşam döngüsü ilerledikçe, modeldeki veriler daha da zenginleşmektedir.

4D YBM Modeli: Yapı Bilgi Modeli için dördüncü boyut, zamandır. Bu boyutta yapı bilgi modeline zamanlama ile ilgili veriler eklenmektedir. Model için zamanla ilgili veriler, teslim süresi, yükleme, kurulum, operasyonel, inşaat yapımı sırasında belirli iş kalemlerinin tamamlanması için gerekli olan sürelerin tespit edilmesi gibi bilgileri içermektedir. Bu bilgiler, projenin gelişimini gösteren doğru program bilgilerini ve görselleştirmelerini elde etmek için kullanılmaktadır. İnşaat planlama çalışmaları söz konusu olduğunda modelde yer alan güvenilir verilere dayalı bir şekilde çalışmak oldukça faydalı olmaktadır. İnşaata başlamadan önce yapının prototipini oluşturabilmek, yapım sırasında karşılaşılabilecek problemlere dair öngörünün

oluşmasına erken bir aşamada izin vermekle birlikte zaman ve para kaybına neden olan revizyonları engellemektedir. Program görselleştirmesi ile proje üyeleri, doğru ve gerçek zamanlı bilginin çoklu kaynaklarına bağlı olarak sağlam kararlar alabilmektedir. Projelerin görsel olarak nasıl inşa edileceğini görmek, paydaşlarla işbirliği yapılırken her bir disipline ait yapılması gereken iş kalemleri hakkında net bir görsel anlayış sağlamaktadır.

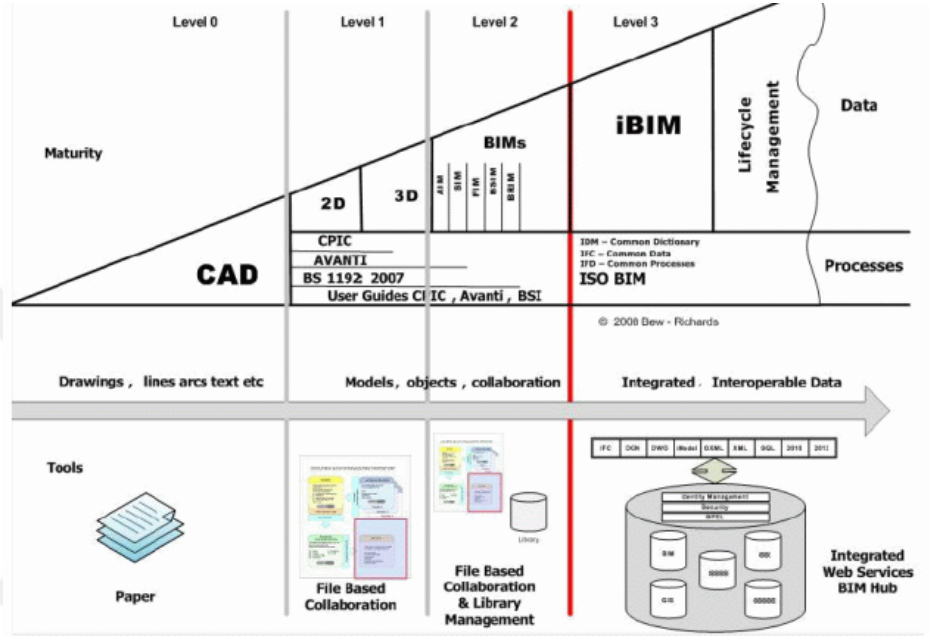
5D YBM Modeli: Beşinci boyut ile modelden doğru maliyet bilgisi çıkartılabilmektedir. Modelde yer alan maliyet ile ilgili veriler, satın alma, yükleme, işletme ve yenilenme ile ilgili maliyet bilgilerini içermektedir. Maliyet hesaplamaları, yapı bilgi modeli içindeki belirli bileşenlere bağlı veriler temelinde yapılmaktadır. Bir modele bağlı maliyet tahmini yaklaşımının faydaları arasında 3D formda maliyetleri kolayca görme, değişiklik yapıldığında geri bildirim alma ve projeye eklenen bileşenlerin / sistemlerin otomatik sayımı bulunmaktadır. Yapı bilgi modelinden maliyet tahminleri yapmanın avantajlarından biri, verilerin proje sırasında herhangi bir zamanda sorgulanabilmesi ve maliyet raporlarını besleyen bilgilerin düzenli olarak güncellenmesidir. 5D yapı bilgi modeli ile oluşturulan maliyet planı, ekiplerin bütçeyi tasarlamalarına yardımcı olmakta ve maliyet yöneticileri, projenin başlangıcından itibaren sürece dahil olabilmektedir. Bu durum projenin erken aşamalarında maliyetlerin daha hızlı ve daha doğru bir şekilde raporlanmasını sağlamaktadır.

6D YBM Modeli: Altıncı boyut enerji hesaplamalarını ve analizlerini içermektedir. 6D ile tasarım sürecinin erken bir aşamasında, binanın daha doğru ve kapsamlı enerji analizleri yapılabilmektedir. Ayrıca binanın kullanım aşamasında binaya yerleştirilen sensörlerden veri toplama, binanın performansını daha iyi anlama ve binanın enerji tüketimini optimize etmeyi amaçlayan bir strateji tanımlamayı da sağlamaktadır.

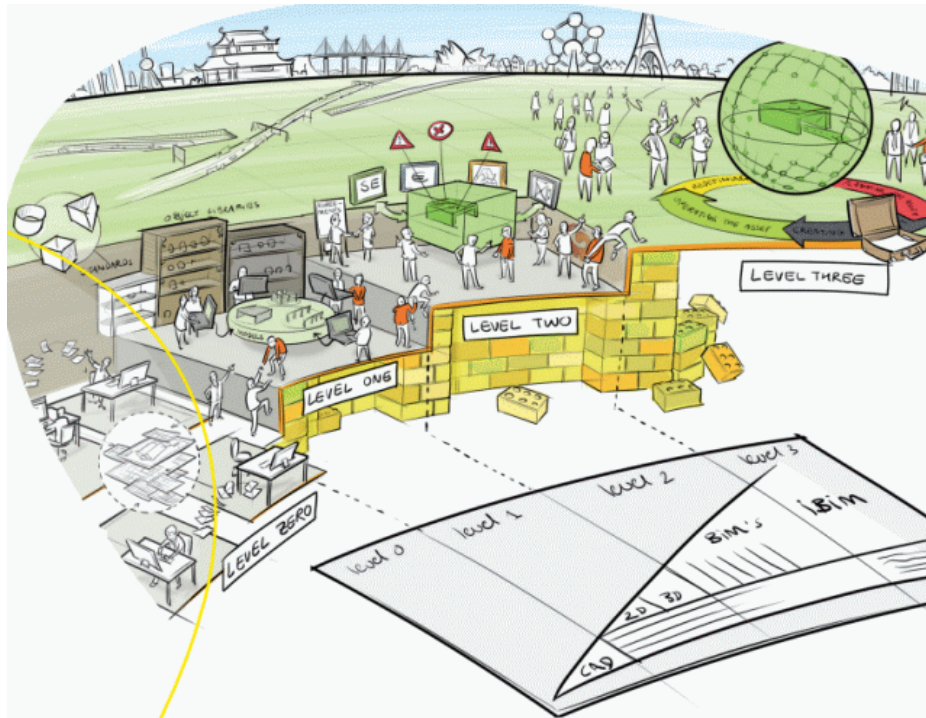
7D YBM Modeli: 7D YBM (yedinci boyutlu yapı bilgi modelleme), yöneticiler tarafından tesisin ömrü boyunca işletme ve bakımında kullanılmaktadır. Bir model oluşturulduğunda ve inşaat aşaması boyunca güncellendiğinde, sahibine devredilebilecek bir “yerleşik” model haline gelme kapasitesine sahiptir. Model, gelecekteki bakım için kullanılabilecek tüm özellik, işletim, bakım kılavuzlarını ve garanti bilgilerini içermektedir.

2.1.3 YBM olgunluk seviyeleri

İstenilen modelleme boyutlarını, destekleyici standartları, rehber notları, bunların birbirleriyle olan ilişkilerini ve sektördeki projelere, sözleşmelere nasıl uygulanabileceklerini açıkça ifade edebilmek amacıyla olgunluk seviyeleri oluşturulmuştur (Şekil 2.4 ve 2.5). Yapılan bu çalışma ile kullanılacak süreçler, araçlar ve teknikler tanımlanmıştır.



Şekil 2.4 : YBM olgunluk seviyeleri (Bew ve Richards, 2008)



Şekil 2.5 : YBM olgunluk seviyeleri (Url-3)

Barnes ve Davies (2015) tarafından YBM olgunluk seviyeleri dört aşamaya ayrılarak aşağıdaki gibi ifade edilmektedir.

Seviye 0: Bu seviyede YBM yer almamaktadır. Tasarım ve üretim bilgileri için 2D CAD dosyaları veya kâğıt kullanılarak veri alışverişi sağlanmaktadır.

Seviye 1: Bu seviye gerçek anlamda YBM için atılan ilk adımdır; tasarım ve üretim için 3D verilerinin kullanılmasını temsil etmektedir. YBM'nin bu seviyesi “yalnız YBM” olarak da adlandırılmaktadır. Bu seviyede kullanıcılar, YBM modeline entegre olmayan maliyet yönetimi gibi bağımsız paketlerden yararlanmaktadırlar.

Seviye 2: Farklı YBM yazılım araçlarının ortak bir modele verilerini ekleyebilmesini sağlamaktadır. Tüm taraflar kendi 3 boyutlu modellerini oluşturmaktadır. Oluşturulan modeller bilgi bakımından oldukça zengin olabilmektedir. Ancak bu seviyede tek bir paylaşılan model üzerinde çalışılmamaktadır. Veriler, ortak bir dosya formatı ile paylaşıldıktan sonra proje iletişimi için kullanılan birleşik bir YBM modeli oluşturmak için bu veriler disiplinler tarafından birleştirilmektedir. Bu YBM seviyesinde, 4D inşaat sıralaması ve 5D maliyet bilgileri kullanabilmektedir.

Seviye 3: Bu seviye, özellikle bilgi teknolojisi konusundaki talepler göz önüne alındığında web veya bulut hizmetleri tarafından kolaylık sağlanması beklenen tam entegre ve işbirliğine dayalı gerçek zamanlı bir proje modeli oluşturmaktadır. Bu seviyede yazılım birlikte çalışabilirliğinin önündeki engeller aşılmaktadır. Tüm disiplinler, merkezde tutulan tek bir paylaşımlı proje modeli aracılığıyla istenilen bilgiye erişebilmektedirler. Tüm taraflar aynı modele gerçek zamanlı olarak erişebilmekte ve değişim yapabilmektedir. Bu durumun avantajı tutarsız bilgilerin oluşmasının önlenmesidir. Bu seviyede, 4D inşaat planlaması, 5D maliyet, 6D sürdürülebilirlik ve 7D proje yaşam döngüsü bilgileri kullanılmaktadır. Aynı zamanda modelde yer alan nesneler için belirli niteliklere sahip kütüphanelerin geliştirilmesi gerekmektedir.

2.2 YBM ve Proje Aşamaları

Projenin yaşam döngüsü aşamalarını oluşturmak, planlama, tasarım, inşaat ve işletme-bakım aşamalarını içermektedir. Her aşama çeşitli proje katılımcıları arasında değiştirilecek birçok bilgiyi kapsamaktadır. Bu bilgi grafiksel ve grafiksel olmayan

verileri içermektedir. Grafiksel veriler çoğunlukla 2D ve 3D çizimler olup, grafiksel veriler diğer proje belgelerini içermektedir. YBM, her aşamadaki tüm bilgileri, kolayca biriktirilebilen, düzenleyebilen ve paylaşabilen tek bir modelde entegre olabilmek için geliştirilmiştir. YBM, işbirlikçi tasarımı ve projenin yaşam döngüsünü kapsayan etkili bir birlikte çalışabilirliği sağlamaktadır.

Geleneksel proje yaşam döngüsü ön tasarım, tasarım, inşaat ve işletme aşamalarından oluşmaktadır. Bu bölümde YBM'nin proje adımlarında nasıl kullanıldığı ele alınmıştır.

2.2.1 Ön tasarım aşamasında YBM kullanımı

YBM sistemlerini konsept aşamasından başlayarak kullanmanın birçok avantajı görülmektedir. Projenin başlangıcında, projenin hedeflerine ve bütçesine dayalı kavramsal bir tasarım oluşturmak mümkündür. Çok disiplinli bir tasarım ekibi kurmadan, projenin fizibilitesini analiz etmek için kavramsal bir tasarım geliştirilebilmektedir. Kavramsal tasarım oluşturulduktan sonra, proje için kaba maliyet tahminleri hesaplanabilmektedir. Projenin fizibilitesine ve maliyetine, konseptine bağlı olarak, proje için herhangi bir ek maliyet olmaksızın tasarım kararları kolayca değiştirilebilirken, birden fazla tasarım alternatifi kolayca oluşturulabilmekte ve karşılaştırılabilmektedir. Analiz ve simülasyon araçlarının şematik model üzerinde kullanımı, binanın gelecekteki kalitesini ve sürdürülebilirliğini artırmaya yardımcı olmaktadır (Eastman ve diğ., 2011).

Projenin ön tasarım aşamasında analiz yapılması, tasarım alternatiflerinin üretilmesi ve karşılaştırılması için YBM sistemi ile entegre çalışan yazılımlar geliştirilmiştir. Bu konu ilerleyen bölümlerde detaylı olarak ele alınmıştır.

2.2.2 Tasarım aşamasında YBM kullanımı

Bir binanın şematik modeli kurulduktan sonra, tasarım aşamasında modeldeki grafik ve grafik olmayan veriler genişletilmektedir. Model, 2D çizimleri dahil etmek yerine doğrudan 3D bilgisine dayanmaktadır. Modelde yapılan değişikliklerin 2D, 3D temsillerine yansıtılmasıyla projenin tamamının gözden geçirilmesi ve revize değişikliklerinin her bir temsile ayrı ayrı uygulanması için harcanan zaman ortadan kaldırılmaktadır.

YBM sisteminin amacı, üretim ve inşaat gibi gelecekteki aşamalar için tüm bilgilerin elde edilebileceği bir 3D temsiline sahip olmaktır. Proje sürecinde üretilmesi gereken 2 boyutlu çizimler dahil bütün veriler 3D modelden alınmaktadır. Böylece, teslim belgelerini oluşturma sürecinde harcanan zaman azaltılarak tasarımcıların tasarım problemine daha fazla odaklanabilmeleri sağlanmaktadır. Projede yer alan farklı disiplinler arasındaki işbirliği bu aşamada oluşturulmaktadır. Bu süreç, farklı disiplinlerin tüm bilgilerini içermekte ve projede karşılaşılan tutarsızlıkları azaltmaktadır (Eastman ve diğ., 2011).

Bu aşamada kullanılan yapı bilgi modelinin veri zengini olması nedeniyle maliyet ve zaman analizleri yapılabilmektedir.

2.2.3 İnşaat aşamasında YBM kullanımı

Bir projenin ilerlemesi sırasında, 3D model içinde barındırdığı verilerle birlikte gelişmektedir. Yapı bilgi modelinde yer alan bilgiler, inşaat aşamasında şantiye planlaması, ürün alımı gibi çeşitli amaçlarla kullanılabilir. Bu aşamada inşaat için gerekli olan farklı bileşenlerin imal edilmesi için model temel alınabilmektedir. Üreticiler, modelde kullanılabilecek ürünlerle ilgili kendi YBM bileşenlerini oluşturabilmektedirler. Böylece, kullanılacak olan bileşenlerle ilgili doğru grafik ve veriler elde edilmektedir. YBM kullanımı şantiye planlaması ile bileşenlerin kurulum süresini hızlandırmakta ve ürünler için gerekli depolama alanını azaltmaktadır. İnşaat aşamasında meydana gelen değişiklikler ve bunların etkisi, yapı bilgi modeli üzerinden daha akıllı ve daha hızlı kararlar vermek için doğrudan analiz edilebilmektedir. Çakışma testleri yapılarak projede olan sorunlar tespit edilebilmektedir. Bu aşamada, önceki proje aşamasında belirtildiği gibi projede görülen tutarsızlıklar önemli ölçüde azaltılmaktadır. Bu, inşaat aşamasında karşılaşılan maliyetleri azaltmakta ve yapım işini hızlandırmaktadır. 4D YBM, inşaat aşamasında kullanılmakta ve inşaat programını modele bağlamaktadır. Yapı bilgi modeli, binanın her geçen gün nasıl inşa edileceğini görselleştirerek, gelecekteki sorunların erken tespitine izin veren bir bilgi kaynağı olarak hizmet vermektedir. Belirli bir zamanda hangi ürünlerin sahada olması gerektiği hakkında bilgi ile ürünlerin sahada depolanması veya zarar görmesi sebebiyle oluşabilecek maliyetlerden kaçınarak doğru bir inşaat planlaması yapılabilmektedir (Eastman ve diğ.,2011).

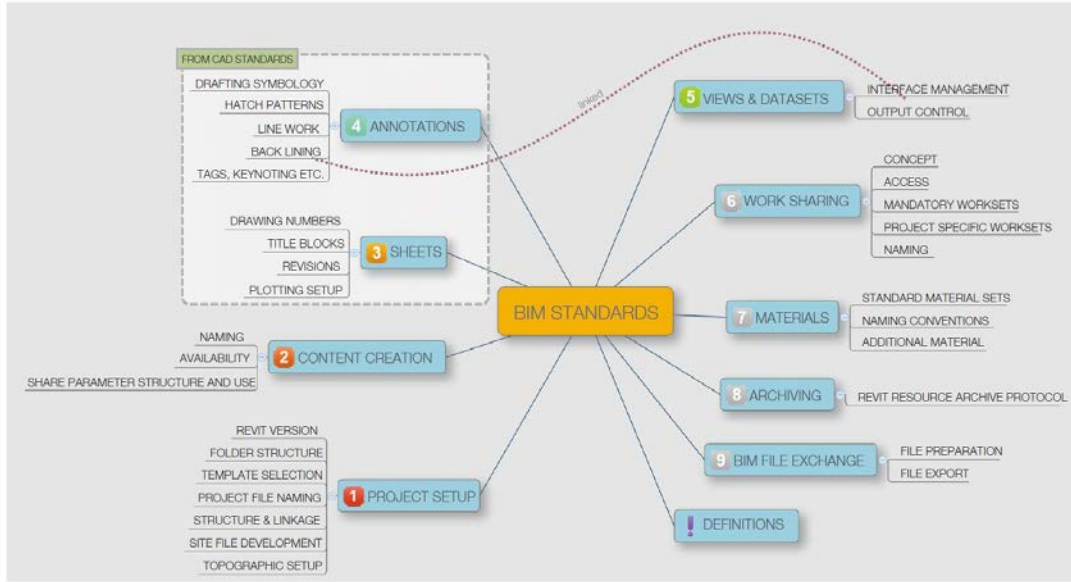
2.2.4 Kullanım aşamasında YBM kullanımı

Projenin teslim sürecinde ilgili disiplinlerce, mimarlık, yapı, sıhhi tesisat, kurulu ekipman gibi konularla ilgili proje sahibine veya işletimini yapacak olan kişilere çok miktarda bilginin sağlanması gerekmektedir. Gerekli tesis yönetim verileri, 3D modele dahil edilebilmektedir. YBM'nin kullanılması, proje yaşam döngüsü aşamalarında gereken inşaat, işletme ve bakım ile ilgili disiplinler arasındaki bilgi transferini kolaylaştırmaktadır. Proje tesliminden sonra, yapı bilgi modelinde toplanan bilgiler operasyonel ve bakım amaçlı kullanılabilir. YBM ve tesis yönetimi arasında bir bağlantı oluşturmak gereklidir. Bu nedenle YBM, kullanılan mevcut tesis yönetim sistemlerine bilgi bağlayan veri formatları ve seçenekleri sunmaktadır. İşletme ve bakım aşaması, projeye ait en uzun dönemdir ve en yüksek maliyetin gerçekleştiği aşamadır. Bir projede YBM'nin uygulanmasının temel amacı, maliyetin azaltılması, verimliliğin ve sürdürülebilirliğin sağlanması açısından bir değer yaratılması ve kullanılmasıdır. Sonuç olarak, projenin kullanım aşamasında YBM'nin gelişime açık bir potansiyeli bulunmaktadır (Eastman ve diğ., 2011).

2.3 YBM Sisteminde Kullanılan Standart ve Formatlar

YBM Standartları, YBM'nin kullanımıyla ilgili kullanıcılar tarafından ortak ve standart bir yaklaşım gerektiren farklı faaliyetlerin bütünüdür (Şekil 2.6). Bu faaliyetler, proje kurulumu, bilgi depolama, bilgi isimlendirme ve bilgi alışverişi protokolleri ile doküman çıktılarının formatı ve diğer teslim edilebilen biçimlerle ilgilidir. Standartlar ayrıca YBM nesnelerinin nasıl kurulduğunu, nasıl adlandırıldığını ve YBM kitaplığı içinde nasıl sınıflandırıldığını da içermektedir.

YBM sisteminin proje teslim sürecine etkin bir şekilde entegre edilebilmesi için, ekibin YBM sistemi için ayrıntılı bir uygulama planı geliştirmesi önemlidir. Bir YBM Uygulama Planının (BIM Execution Plan) genel vizyonu, proje ekibi için proje boyunca takip edilecek uygulama detayları ile birlikte sürecin ortaya koyulmasıdır. YBM Uygulama Planı bir projenin ilk aşamalarında geliştirilmelidir; projeye ek katılımcılar eklendikçe de geliştirilmeye devam edilmelidir. Projenin uygulama aşamasında ihtiyaç duyulduğu anda revize edilebilir. Plan, projede YBM uygulamasının kapsamını tanımlamalı, YBM görevleri için süreç akışını tanımlamalı, taraflar arasındaki bilgi alışverişini tanımlamalı ve uygulamayı desteklemek için gerekli olan proje ve şirket altyapısını tanımlamalıdır (Penn State, 2011).



Şekil 2.6 : YBM Standartları (Holzer, 2016)

Günümüzde AIA (American Institute of Architects), IAI (International Alliance for Interoperability), NIBS (National Institute of Building Sciences), endüstri uzmanları, teknoloji şirketleri gibi birçok organizasyonun da katkısı sonucu YBM sisteminde standartlaşmış bir üretim süreci tanımlanmıştır.

Uluslararası standartlar, bir ürünün herhangi bir sistemden bağımsız olarak tüm yaşam döngüsü boyunca veri tanımının yapılabilmesini sağlamak için oluşturulmaktadır. YBM sisteminde görülen veri değişiminin uygulamadan bağımsız bir yapı kazanması ortak kabul edilen standartlara bağlı olmaktadır. YBM sistemini kullanan farklı yazılımların veri kaybı olmadan birlikte çalışabilmesi önemlidir. Bu konuda pek çok çözüm önerisi oluşturulmuştur. Bunlardan bazıları Uluslararası Standart Organizasyonu (ISO)'nun geliştirdiği STEP'dir. 1980'lerden itibaren ISO, ürün model verisi için standartlaşmaya gidilmesi amacıyla STEP'i geliştirmiştir. STEP kısaltmasıyla kullanılan Standart for the Exchange of Product Model Data, bir ürünün yaşam döngüsü boyunca sahip olduğu tüm fiziksel ve işlevsel özelliklerinin uluslararası ürün veri standardıdır. STEP, oldukça geniş ve kapsamlı bir içeriğe sahiptir. STEP, bir veya daha fazla ürün bilgisini temsil etmek için Uygulama Protokollerini (Application Protocols) kullanmaktadır (SCRA, 2006).

Genel ürün veri modeli değişim biçimi olan IFC (Industry Foundation Class) açık-standart yapı modellerini içermektedir. Ürün veri modeli biçimleri, geometrinin yanı sıra nesne, malzeme özellikleri ve nesneler arası ilişkilerin de tanımlanmasını

sağlamaktadır ki, bunlar analiz ve inşaat yönetimi gibi konularda gerekli olan verileri barındırmaktadır (Eastman ve diğ., 2011).

IFC (Industry Foundation Classes), herhangi bir yazılım sağlayıcının kontrolünde olmayan, ISO-STEP EXPRESS diliyle oluşturulmuş, açık bir veri modelidir. International Alliance for Interoperability (IAI) tarafından 1995 yılında geliştirilen veri modellenli bir nesne yönelimli dosya biçimidir. İnşaat endüstrisinde ve özellikle YBM alanında birlikte çalışabilirliğin ve yazılımdan bağımsız veri değişiminin sağlanması için kullanılmaktadır. ISO tarafından uluslararası düzeyde kabul görmüş ve ISO/PAS 16739 olarak sınıflandırılmıştır (Eastman ve diğ., 2011).

YBM ile yürütölen bir projede bilginin standardizasyonu, projede üretilen ve yönetilen bilginin düzgün bir şekilde kullanılması ve aktarılması için gerekli olmaktadır. Üretilen bir model hakkında bilgi sahibi olmak yeterli olmamaktadır. Modelde yer alan bilginin evrensel olarak anlaşılabilir ve erişilebilir olması gerekmektedir. Aksi takdirde, modeldeki veriler bilgileri ilk olarak modele ekleyen kişiler için yararlı olup diğör kullanıcılar tarafından anlaşılmaz olabilmektedir. Bu nedenle tasarım ve inşaat endüstrisi tarafından ortak kullanılan standartlar ve formatlar bulunmaktadır. Bu formatlar yapılan iş bölümlerini, proje içerisinde yer alan yapı elemanlarının çeşitlerini ve seçilen tekil yapı ürünlerini ele almaktadır. Bilginin doğru bir şekilde modele işlenebilmesi için belirli sınıflandırma sistemlerine ihtiyaç duyulmaktadır.

Mimarlık, mühendislik ve inşaat sektöründe sınıflandırma sistemlerine ihtiyaç duyulmasının en önemli sebeplerinden bir tanesi proje üretim sürecinin büyük bir işbirliği gerektirmesinden kaynaklanmaktadır. YBM ile işbirliği, çok sayıda insan arasında uzun süreler boyunca geniş bilgi alışverişini gerektirir. Her katılımcı rollerine bağılı olarak, proje üretim döngüsünün farklı zamanlarında farklı bilgi ihtiyacına ve sorumluluklarına sahiptir. Projenin ölçeğı ne kadar büyük olursa ve katılımcı sayısı ne kadar artarsa, bilgi alışverişı ile ilgili karmaşıklıkları idare edebilen yöntem ve sistemlerin kullanması bir o kadar önemli olmaktadır. Bütün katılımcılar tarafından üzerinde anlaşmaya varılan bilgilerin tutarlı bir şekilde sınıflandırılması, amaçların iletilmesi, yanlış anlama ve çatışmanın azaltılması önemlidir. Çünkü projeye dahil olan bireyler projeden projeye değişebilmektedir.

2.3.1 YBM Sınıflandırma Türleri

Sınıflandırma sistemleri ile inşaat sektöründe farklı açılardan kullanılabilecek terminoloji ve semantik oluşturulmaktadır (Ekholm ve Häggström, 2011). Bu durum sistem içerisinde mevcut olan ve sisteme ilave edilecek olan bilgileri özetlemeye ve düzenlemeye yardımcı olmaktadır. İnşaat endüstrisinde, sınıflandırma sistemlerini kullanmak, şartname ve diğer belgelerin yapılandırılmasında, maliyetlerin hesaplanmasında ve paydaşlar arasında veri paylaşımında bulunulması gibi konularda kritik önem taşımaktadır (Ekholm, 1996). Daha önemlisi, YBM paradigmasında bina ürün modellerini standart bir şekilde sınıflandırma, ürün modellerinin düzenlenmesindeki en önemli adımların başında gelmektedir (Eastman ve diğ., 2011). Ürün modellerine uygun sınıflandırma kodu verilerek, bina modeli içinde inşaat bilgisi veya maliyet tahmini için ayarlanabilmekte ve ayrıca ürün veri tabanları içinde sıralanabilmektedirler.

Sınıflandırma, inşaat birimlerini standart bir şekilde tanımlamak için bir araçtır. Bir sınıflandırma sistemi, bir dizi nesneyi belirli özelliklere sahip üyelerden oluşan farklı sınıflara ayırmaktadır (Ekholm, 1996). Bir sınıf, bir veya daha fazla ortak özelliği olan bilgi nesneleri koleksiyonuna atıfta bulunan kavramsal bir yapıdır. Sınıflandırma yapmak, nesnelerin koleksiyonunu birbirinden ayrılmış kümeler ya da sınıflar halinde bölmek anlamına gelmektedir (Ekholm ve Häggström, 2011).

Nesnelerin ve yapılan işlerin sınıflandırılmasında ilk kritik adım, sınıflandırmanın amacını tanımlamaktır (Ekholm, 1996). İyi organize edilmiş bir yapı sınıflandırma sistemi, bir projenin yaşam döngüsü boyunca bir bilgi merkezi olmasında önemli bir rol oynamaktadır. Ulusal sistemlerin standarttaki aynı sınıf tanımlarını izlemesi durumunda yapılan sınıflandırma, daha kolay takip edilebilmektedir. Böylece modele ait temel kategoriler hakkında bir bilgi sağlanmaktadır. Örneğin; ISO standartları, kuruluşlar tarafından ulusal ya da bölgesel bazda kullanılabilen bina sınıflandırma sistemlerini geliştirmek için temel alınmıştır. ISO sınıflandırma kapsamı, inşaat yönetimi ve ürünlerini içeren yaşam döngüsü bilgilerine sahip olacak şekilde geliştirilerek birçok bina sınıflandırma sistemi için kullanılmıştır (Ekholm ve Häggström, 2011).

Yapı sınıflandırma sistemleri, nesnelerin gruplandırılması için iki farklı ilkeye sahiptir. Bu gruplandırma stratejileri doğrudan gruplandırma (numaralandırılmış, hiyerarşik gruplandırma) ve kombinasyonel, fasetli gruplandırma stratejisi olabilmektedir.

Günümüzde kullanılan başlıca sınıflandırma sistemlerinden olan MasterFormat, UniFormat, OmniClass ve UniClass bu bölümde ele alınmıştır. Her bir sistem farklı amaçlar için geliştirilmiştir.

MasterFormat: MasterFormat, CSI ve CSC tarafından üretilmektedir (MeLampy, 2007). Kuzey Amerika'daki çoğu inşaat projeleri için kullanılmaktadır ve 1960'lardan bu yana inşaat bilgilerinin düzenlenmesi için bir standart olmuştur (CSI ve CSC, 2004). MasterFormat, inşaat çalışmaları sonuçlarını, gereksinimlerini, ürünlerini ve faaliyetlerini organize etmek için kullanılan iş sonuçlarına göre numaralar ve başlıklardan oluşan bir sınıflandırma sistemidir. Birincil kullanımı, teklif verme ve sözleşme gereklilikleri, şartnameler ve ürün bilgilerini düzenleme konusunda olmuştur (MeLampy, 2007).

Her MasterFormat numarası ve başlığı, düzeylerde düzenlenmiş bir bölüm ile tanımlanmaktadır. Her bölüm, seviye 2, seviye 3 ve çoğunlukla dört rakamdan oluşur ve başlıklar açıldıkça daha ayrıntılı alanlar belirtilmektedir. Bu bölümlenme ve numaralandırma sistemi CSI tarafından gelecekte beklenen büyüme ve genişlemeye göre yapılandırılmıştır (CSI ve CSC, 2004).

UniFormat: Bir maliyet danışmanı olan Hanscomb Associates, 1973'te Amerikan Mimarlık Enstitüsü (AIA) için Mastercost adlı bir sistem geliştirirken, ABD Genel Hizmetler Yönetimi de (General Services Administration-GSA) bir sistem geliştirmektedir. AIA ve GSA, ortak bir sistem üzerinde anlaşmaya varmış ve sistemi UniFormat olarak adlandırmıştır. Daha sonra UniFormat'a dayalı yapı unsurlarının sınıflandırılması için UniFormat II adında bir standart geliştirilmiştir (Charette ve Marshall, 1999). Uniformat'ın en son sürümü 2010 yılında yayınlanmıştır.

MasterFormat, bilgileri iş sonuçlarına göre sınıflandırırken, UniFormat bilgileri bir projedeki yapı fonksiyonel elemanlarına göre sınıflandırmaktadır. Yapı elemanları olarak bilinen sistemlerin ve montajların fiziki parçaları çevresindeki yapı bilgilerini düzenlemektedir. Bu sistemler ve montajlar, çalışma sonucunu tanımlamadan işlevleriyle karakterize edilmektedir. Temel kullanımı, tasarımcılarının şematik tasarım aşamasından başlayarak maliyet tahminleri sunabilmesidir. UniFormat, özel

işlevleriyle ayrılmış önemli yapıım kategorilerini barındırmaktadır. Bu işlevler beş hiyerarşik seviyedeki dokuz kategoriye içermektedir (Charette ve Marshall, 1999). A: Altyapı, B: Kabuk, C: İç Mekanlar, D: Hizmetler, E: Ekipman ve Mobilyalar, F: İnşaat ve Yıkım, G: Bina Saha Çalışması ve Z: Genel

UniClass: Uniclass, 1997 yılında İngiltere'de yayınlanmıştır. Uniclass, tasarım ve yapıım sürecinin tüm yönleri için bir sınıflandırma sistemi olarak geliştirilmiştir. Kütüphane materyallerinin düzenlenmesi, ürün modellerinin ve proje bilgilerinin yapılandırılması için tasarlanmıştır (Kang ve Paulson, 2000).

Uniclass sınıflandırma sisteminde yer alan tablolar benzer varlıkları gruplandırırken, hiyerarşisi iki basamaklı bir koddan altı basamaklı bir koda kadar genişleyebilmektedir. Tablo J "yapılar için iş bölümleri", tablo K "inşaat mühendisliği çalışması", tablo G "bina elemanları" ve tablo L "inşaat ürünleri" için kullanılmaktadır.

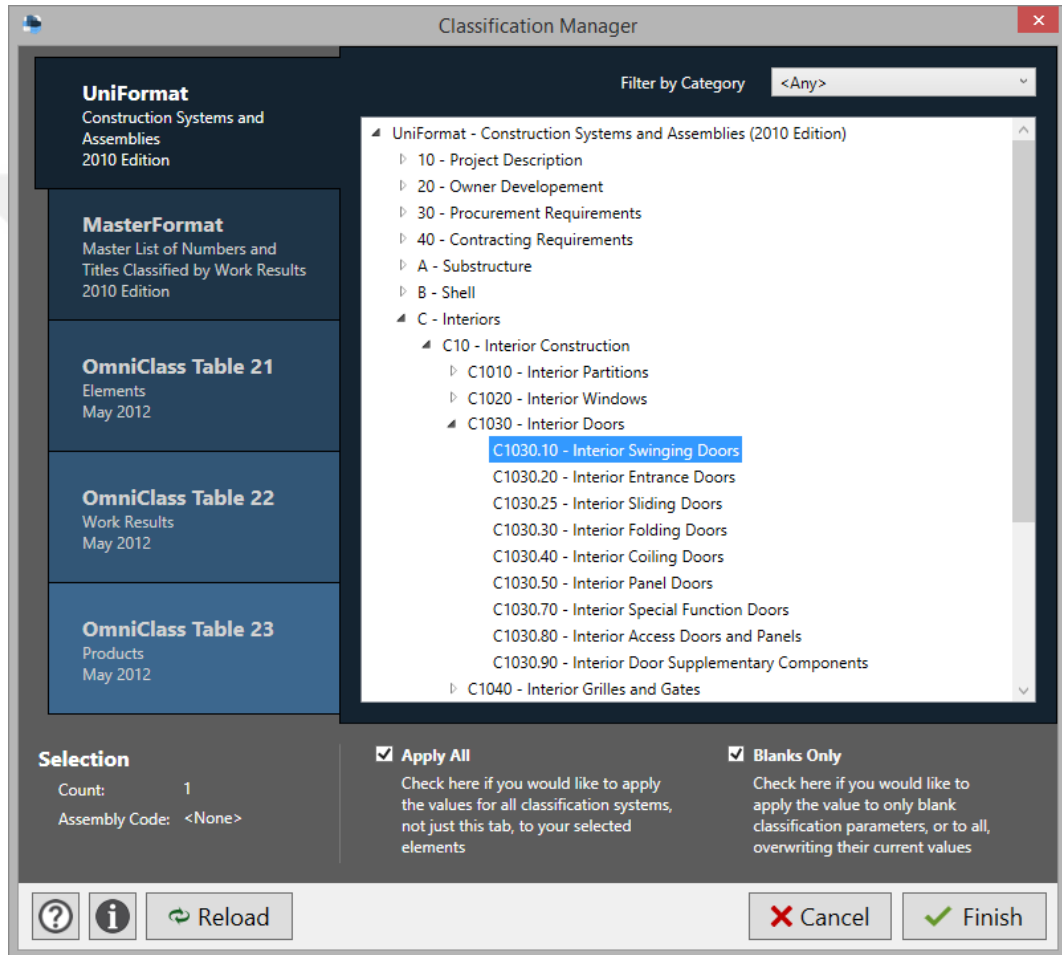
OmniClass: OmniClass sınıflandırma sistemi tüm inşaat bilgilerini organize etmek için kurulmuştur. OmniClass'ın önemi Amerika'da Ulusal Yapı Bilgi Modelleme standardı için kullanılmasıdır (Brodt, 2016). OmniClass, proje başlangıç aşamasından teslim aşamasına kadar olan süreçlerdeki organizasyona yardımcı olunması, maliyet ve şartname bilgilerinin sıralanması, projeden istenen bilginin elde edilmesi ve bu bilgilerin tesis yönetimi için kullanılması üzerine tasarlanmıştır. Proje kütüphanesi içerisinde oluşturulan eleman ve malzemeleri düzenleme gibi birçok uygulama için de yararlıdır. OmniClass, iş sonuçları tablosu (tablo 22) için MasterFormat'ı, elemanlar tablosu (tablo 21) için UniFormat'ı baz almaktadır.

İnşaat sektöründe bilgi teknolojilerinin hızla gelişmesi ve inşaat malzemelerinin küreselleşmesi, standardizasyonu ve sınıflandırma sistemlerinin uluslararası düzeyde koordinasyonunu gerektirmektedir. Sınıflandırma sistemleri yeni üretilecek olan yapı elemanları ve süreçlere de karşılık verebilmektedir.

Yapı Bilgi Modelleme sistemleri kullanılırken en çok zaman kaybı yaratan işlerin başında, sisteme yüklenmesi gereken verinin YBM standartlarına uygun bir şekilde modele ve sürece dahil edilebilmesi gelmektedir. Bu nedenle verinin doğru bir şekilde modele işlenebilmesi için sınıflandırma sistemlerine ihtiyaç duyulmaktadır. Verinin YBM sistemi içerisinde koordinasyon, üretim gibi konularda işlenerek tesis yöneticiliğine kadar doğru bir şekilde ulaştırılması için YBM projesine başlamadan önce dikkat edilmesi gereken en önemli noktalardan birisi de kullanılacak olan

sınıflandırma sistemine karar verilmesidir. Sınıflandırma sistemi, yapı bilgi modeli yaşam döngüsü boyunca MMİ endüstrisindeki işbirliği için oldukça önemli olmaktadır.

Sınıflandırma sistemlerinin modelde yönetilebilmesi için YBM yazılımlarına eklentiler geliştirilmiştir. Autodesk, “Classification Manager” eklentisini (Şekil 2.7) sınıflandırma sistemlerinin YBM sistemine kolay bir şekilde entegre olabilmesi için geliştirmiştir.



Şekil 2.7 : Autodesk Classification Manager eklentisi

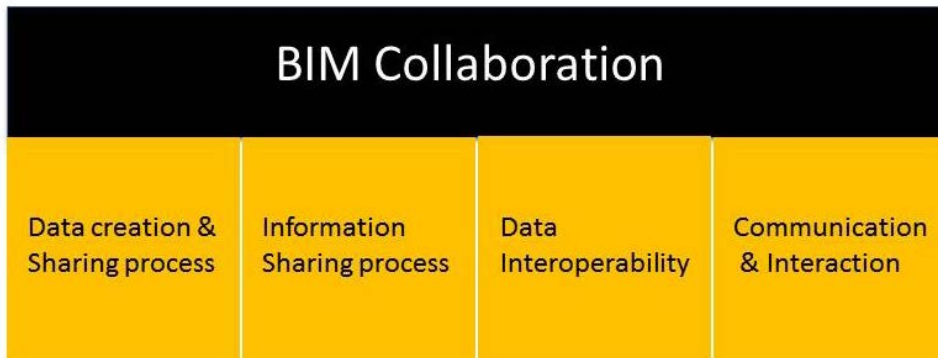
Sonuç olarak sınıflandırma sistemlerine, modeldeki verinin anlaşılabilir olması, yapılacak olan analizlerin ayrıntılı ve doğru olması, projeye dahil edilen disiplinler arasında standardizasyonun olması, proje maliyet tahminleri ile ihale dosyalarının oluşturulması, üreticiye ürün bilgisinin aktarılması ve modelde yer alan verilerin tesis yönetimi kapsamında kullanılabilmesi için ihtiyaç duyulmaktadır.

2.4 YBM ve İşbirliği

Yapı sektörünün parçalanmış yapısı, karmaşıklığı ve inşaat yaşam döngüsünün birçok aşaması, çok disiplinli ekiplerin (proje sahipleri, mimarlar, danışmanlar, mühendisler, yükleniciler, taşeronlar, tedarikçiler, vb.) işbirliğinin önemli olduğunu göstermektedir.

YBM teknolojilerinin proje işbirliğinde etkin olarak kullanıldığı tümleşik proje uygulamalarında, paydaşlar, ortak ürünleri olan binayı sistemsel bir mantıkla üretmekte ve projenin farklı evrelerinde birbirleriyle dosya alışverişinde bulunmaktadır. Paydaşların karşılıklı bilgi değişimine dayalı iletişimine birlikte çalışabilirlik (interoperability) adı verilmektedir (Ofloğlu, 2008).

YBM işbirliği, belirli bir proje için farklı YBM aşamalarının tanımlanması ve bu farklı aşamalar arasında veri ve bilgi paylaşımının tanımlanması sürecidir. Bu durumda, “veri”, YBM projesinin farklı aşamalarında kullanılan farklı yazılımlar tarafından üretilen verilere atıfta bulunur ve “bilgi”, YBM projesinin farklı aşamalarında kullanıcıları bilgilendirme sürecini ifade eder. YBM işbirliği süreci, üzerinde düşünülmesi gereken pek çok şeyle birlikte karmaşık görünebilir; veri oluşturma, kullanılacak yazılım araçları, veri yönetimi çözümleri, bilgi paylaşımı, veri kopyaları vb. Bu nedenle, YBM işbirliğini Şekil 2.8’de gösterildiği gibi farklı bileşenlere ayırmak mantıklıdır (Koppula, 2012).



Şekil 2.8 : YBM işbirliği bileşenleri (Koppula, 2012)

Veri oluşturma ve paylaşma süreci, verileri tanımlamak ve düzenlemek, veri yönetimi çözümünü geliştirmek ve verileri paylaşmak için iş akışlarını tanımlamakla ilgilidir. Bilgi paylaşımı süreci, bilgilerin aynı tasarım disiplini ve diğer tasarım disiplinleri ile paylaşıldığı farklı aşamaları tanımlamayı, paylaşılan verilerin durumunu ve sürüm bilgilerini vb. içermektedir. Veri birlikte çalışabilirliği, projede yer alan farklı veri formatlarıyla ilgilenmekte ve gerekli olabilecek veri geçişleri de dahil olmak üzere

bunları bir araya getirmek için iş akışlarını tanımlamaktadır. İletişim ve etkileşim, kullanıcıların projenin en son durumundan ve tasarım verilerinden haberdar oldukları bildirim ve iletişim süreci ile ilgilidir. Bunlar, iki yönlü işbirliği için gereken gerekli yazılım ve donanımlarının kullanımı gibi konuları içerebilmektedir (Koppula, 2012).

Mimarlık, mühendislik ve inşaat sektöründe disiplinler arası koordinasyon eksiklerinden kaynaklı ortaya çıkan revizyonlar nedeniyle projelerin yapım süreçlerinde zaman ve maliyet kayıpları yaşanmaktadır. Bu anlamda sektörde görülen koordinasyon eksikliğinin sebeplerinin başında aynı projede çalışan farklı disiplinlerin birbirleri ile bilgi paylaşımı ve iletişim eksikliğinin olmasıdır. YBM sisteminde proje ile ilgili disiplinler ortak bir sanal model üzerinde çalışmaktadırlar. Bu sistem ile koordinasyon eksikleri en aza indirilmektedir. Böylece disiplinler arasındaki iletişim ve bilgi paylaşımına bağlı olarak koordinasyon sağlanmış olmaktadır.

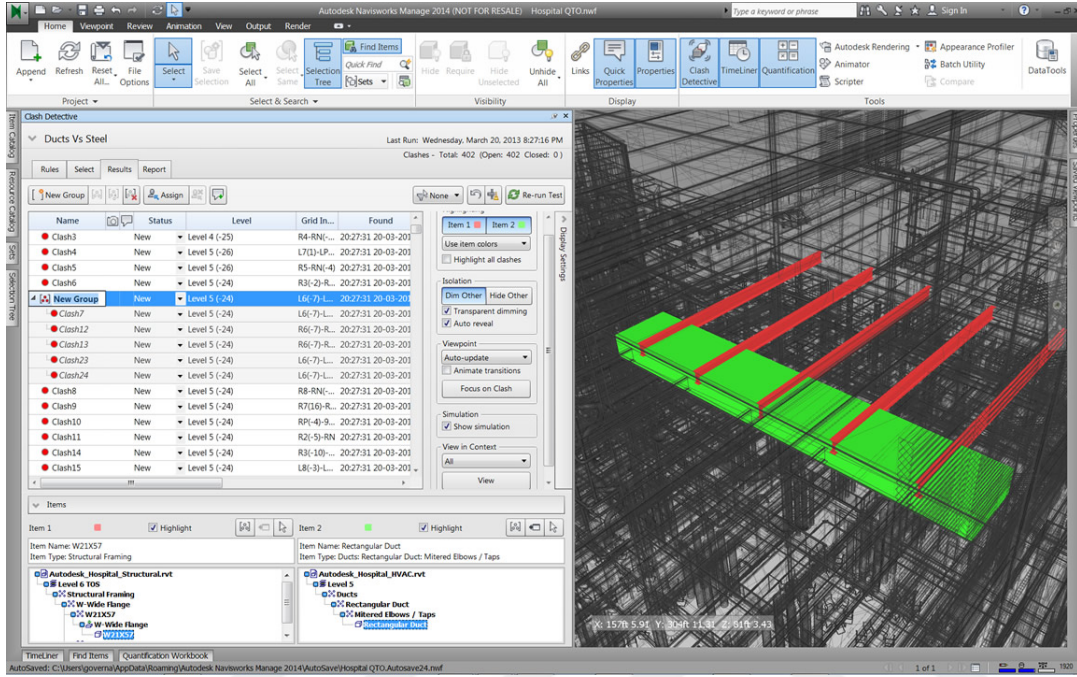
Yapı Bilgi Modelleme, bina ile ilgili grafik (geometri/biçim vb.) ve alfasayısal (malzeme, maliyet, fiziksel çevre kontrolü vb.) veriden oluşan üç boyutlu bir model meydana getirerek, bu modelin yapı sektörü paydaşları tarafından ortak kullanımını sağlayan bir çalışma yaklaşımıdır (Şekil 2.9). Bu üç boyutlu model, planlama, tasarım, projelendirilme, yapım ve işletim gibi projenin tüm yaşam döngüsünü içeren süreçlerinde kullanılabilmektedir. Farklı paydaşların aynı modeli kullanabilmesi temsilde tutarlılığı arttırmakta, revizyon kolaylığı sağlamakta; veri dönüştürme işlemlerini, verinin tekrarlı üretimini ve proje belgeleri arasında ilave ilişkilendirme veya koordinasyon ihtiyaçlarını önemli ölçüde azaltmaktadır (Ofloğlu, 2008).



Şekil 2.9 : Mimari, statik ve mekanik model (Url-4)

Yapı bilgi modeli üzerinde yapılacak olan koordinasyon çalışmaları ilk aşamada görsel olarak yapılabileceği gibi proje sürecinde YBM tabanlı yazılım araçlarından da

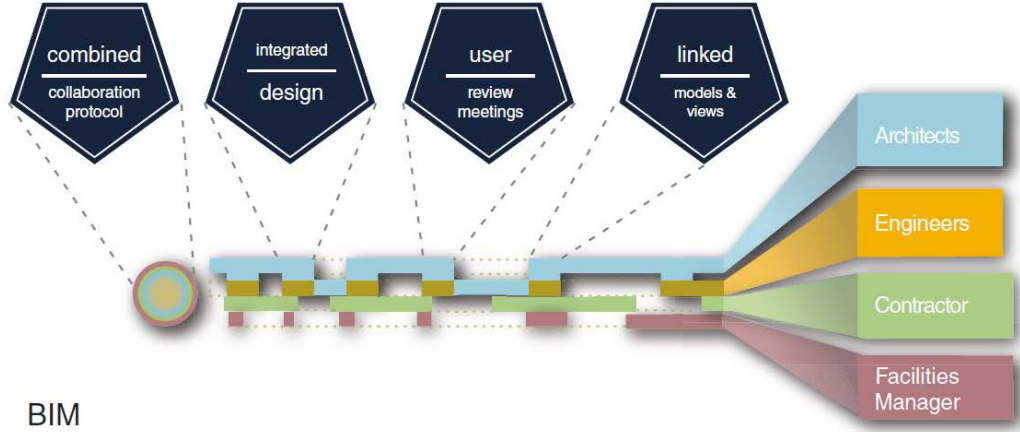
yararlanılmaktadır. Örneğin, Autodesk Navisworks yazılım aracı ile farklı disiplinlere ait modeller arasında veya her disiplinin modeline ait nesneler ile ilgili çakışma testleri yapılabilmektedir (Şekil 2.10). Çakışma testleri, yapı bilgi modeli belli bir seviyeye ulaştıktan sonra değil modellemenin her aşamasında yapılabilmektedir.



Şekil 2.10 : Navisworks programı ile yapılan çakışma testleri (Url-4)

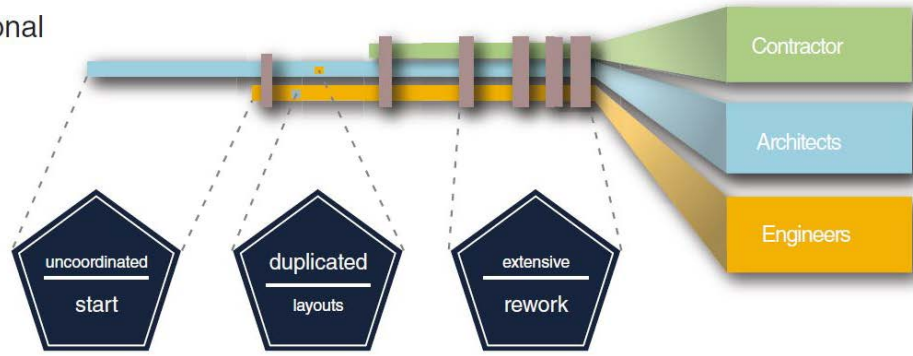
Gerekli bilgileri içeren üç boyutlu görsel bir model üzerinde bilgi alışverişi yapılması ile üretilmiş olan herhangi bir bilginin kaybolmasının ve iki boyutlu projeler üzerinde yapılan çalışmalarda oluşan yanlış anlaşılmanın önüne geçilebilmektedir. Örneğin başka departman tarafından üretilen bir statik yapı modeli üzerinde çalışan mekanik departman, herhangi bir kanal çizimi yaparken statik modeli görerek çalışabileceği için kiriş içinden geçen bir kanal çizimi yapılmayacaktır. Böylece disiplinler arası sağlanan koordinasyon ile birlikte projenin ilerleyen aşamalarında karşılaşılabilecek problemler modelleme aşamasında çözülebilmektedir.

Geleneksel işbirliği ve koordinasyon uygulamasında her disiplin, kendi modellerini lineer bir dizi içinde paylaşmaktadır. Herhangi bir zamanda, ekip üyeleri modelin farklı ve yerel olarak saklanan bir versiyonu üzerinde çalışabilmektedir. Bu yaklaşım, verimsizlik oluşturmakta ve modelin farklı sürümleri ile içindeki bilgilerin eşzamanlılığını giderek kaybetmesine sebep olduğu için tüm tarafları daha fazla risk altında bırakma potansiyeline sahip olmaktadır (Şekil 2.11).



BIM

Traditional



Şekil 2.11 : YBM ve geleneksel işbirliği sürecinin karşılaştırılması
(Holzer,2008)

Son yıllarda, bulut tabanlı işbirliği yeni bir koordinasyon stratejisi olarak karşımıza çıkmaktadır. Bu strateji, çoklu paydaşları ortak bir platformda bir araya getirerek fikir birliğine dayalı karar almayı amaçlamaktadır. Bulut teknolojileri, yüksek performans yetenekleri, erişilebilirlik ve ölçeklenebilir depolama kapasiteleri nedeniyle YBM tabanlı işbirliğinin geliştirilmesine yardımcı olmaktadır. Daha yüksek düzeyde işbirliği oluşturmakta ve proje üyeleri için etkili, gerçek zamanlı iletişim platformu sağlamaktadır.

Autodesk, hem sahada hem de saha dışında proje koordinasyonunu geliştirmek için YBM sistemlerini bulut bağlantılı cihazlara uygulamaktadır. Autodesk'in geliştirdiği BIM 360 Glue, tüm proje ekibini bir araya getiren, zaman ve konumdan bağımsız olarak güncel proje verileri ile modellerine erişim sağlayan bir bulut tabanlı YBM yönetim ve işbirliği hizmetidir. Bulut bağlantısı sayesinde kullanıcılar, ofiste veya sahada mobil cihazlar ile BIM 360 Glue uygulamasını kullanarak YBM modellerine ve akıllı nesne verilerine erişebilmektedirler (Url-5).

Bir BCF dosyası, modeldeki sorunlar, bu sorunların durumu, sorunların atandığı sorumlu kişi, önerilen eylemler ve eylemlerin yapılıp yapılmadığı ile ilgili güncel durumu hakkında bilgi içermektedir. BCF Manager (Şekil 2.12) ile modeldeki sorunlara dair oluşturulan konu ile ilgili metin, modelden ekran görüntüleri ve daha fazlası farklı disiplinler arasında paylaşılmaktadır. BCF’de oluşturulmuş sorunlar doğrudan modeldeki konuma ulaşılmasını sağlamaktadır. Bu durum koordinasyon için harcanan zamanı minimize etmektedir. BCF'nin kullanımı ile büyük YBM dosyalarının internet üzerinden aktarılma ihtiyacı da ortadan kalkmaktadır.

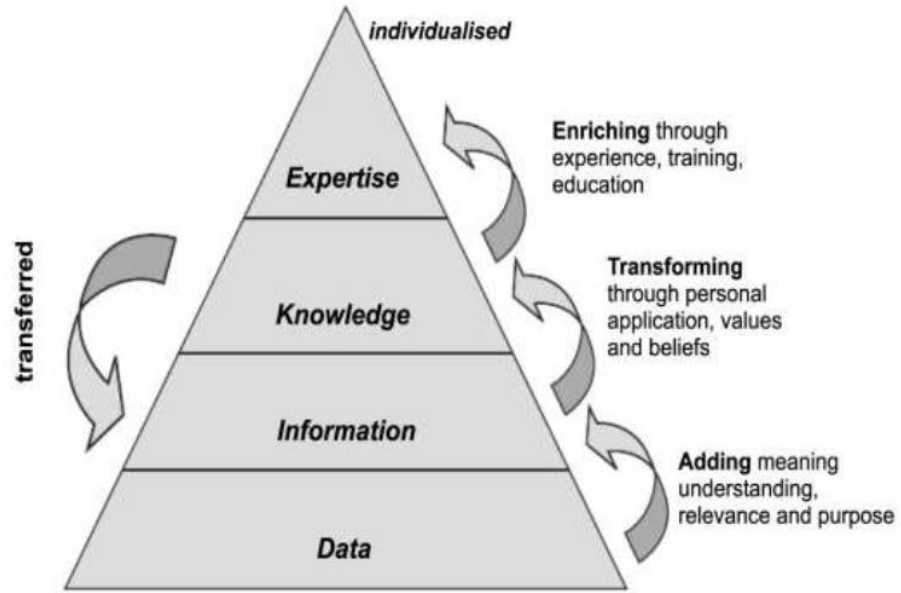


Sonuç olarak, bulut çözümlerinin YBM tabanlı işbirliği sürecinde kullanılması, iş akışlarının proje planlama, tasarım, inşaat ve işletme yaşam döngüsü boyunca entegre edildiği ve iletişimin önündeki engellerin kaldırıldığı ve böylece proje işbirliğinin gerçek zamanlı olarak gerçekleşebileceği anlamına gelmektedir.

2.5 YBM ve Veri

2.5.1 Veri, enformasyon, bilgi (data, information, knowledge)

Veri (data), birbiri ile ilişkilendirilmemiş ham kayıtlardır. Enformasyon (information), işlenmiş veya anlam kazanmış verilerdir. Bilgi (knowledge) ise, değer kazanmış enformasyondur ve enformasyonun amaca yönelik olarak bir araya getirilmesidir (Myers, 1996). Veri, enformasyon ve bilgi kavramları birbirleriyle yakından ilişkilidir. Bilginin enformasyondan daha yüksek bir seviyede olduğu ve verilerin enformasyondan daha yüksek bir seviyede olduğu yaygın olarak bilinmektedir (Bender ve Fish, 2000). Şekil 2.13’de bilgi hiyerarşisi grafiği yer almaktadır.



Şekil 2.13 : Bilgi Hiyerarşisi (Bender ve Fish, 2000)

Veri: Davenport ve Prusak (2000) verilerin, olaylar hakkında objektif olguların bir kümesi olduğunu ve hiçbir yargılama, yorumlamaya dair bir eylem temeli sağlamadığını belirtmişlerdir. Veri, sözdizimsel varlıklar ve anlamsız kalıplardır ve varlığın ötesinde herhangi bir anlam taşımayan kullanılabilir veya kullanılmayan formlarda bulunurlar.

Enformasyon: Enformasyon, anlamları olan yapılandırılmış verilerdir ve verilerin yorumlanmasından üretilir (Davenport ve Prusak 2000). Ackoff (1990), enformasyonu yararlı olarak işlenmiş, kim, ne, nerede ve ne zaman sorularına cevaplar sağlayan veri olarak tanımlamıştır.

Bilgi: Bilgi bugün şirketler için en önemli kaynak ve varlık haline gelmiştir. Bilgi, toplumdaki bireysel ve organizasyon etkileşimleri tarafından oluşturulmuştur. Bilgi, bilişsel çabalarla oluşturulmasıyla birlikte enformasyon ve verilere göre yargılamalar içermektedir (Tuomi, 1999). Özetle, bilginin rolü, verilerin yorumlanması yoluyla verilerin enformasyona dönüştürülme süreçlerine olanak tanımak, var olandan detaylı araştırma ve öğrenme yoluyla yeni bilgi edinme süreçlerini ortaya çıkarmaktır (Aamodt ve Nygård, 1995).

2.5.2 Mimarlık, mühendislik ve inşaat sektöründe veri

21. yüzyıl boyunca meydana gelen teknolojik ilerlemeler ile birlikte yapılı çevremiz ve içinde yaşayan insanlar hakkında büyük oranda bir veri artışı olmuştur. Günümüzde veriler, sensörler, akıllı sayaçlar, sosyal medya ve cep telefonları gibi teknolojilerle üretilebilmektedir. Üretilen büyük hacimdeki veriler, büyük veri (big data) olarak adlandırılmaktadır. Büyük verilerin ve yapılandırılmamış veri kümelerinin sektörler için kullanışlı hale gelebilmesi için zekice analiz edilmesi gerekmektedir. Veri setleri, birçok endüstri alanında yenilikçi yollar ile analiz edilerek maliyetten tasarruf etmek ve ortaya çıkan ürünleri iyileştirmek için kullanılmaya başlanmıştır. Ulaşım, otomotiv, endüstriyel, kamu hizmetleri ve perakende sektörlerinde dünya çapında kullanılan ağ bağlantılı sensör düğümleri vardır. Bu sensörlerden elde edilen veriler sektör içerisinde kullanılmaktadır. Oysa, planlama ve tasarım süreçleri için bu tür verilerin ve yeni teknolojilerin kullanılmasının avantajlarını keşfetmek için diğer endüstrilere kıyasla çok az şey yapılmıştır. Diğer endüstri alanlarında edinilen avantajlar ile büyük veriler, mimari tasarım sürecini ve mevcut ürünü iyileştirmek için büyük potansiyele sahip olabilmektedir. Veri, mimarları tasarım sürecine başlamadan önce tasarımın yapılacağı alan ve yapıyı kullanacak kullanıcı ihtiyaçları hakkında bilgilendirmek için yeni imkânlar sağlamaktadır. Veri aynı zamanda tasarım sürecini hızlandıracak, geliştirecek bir fırsat sunmaktadır. Bu faydalar, daha sürdürülebilir biçimde gelişen ve daha verimli çalışacak kentsel alanları oluşturabilmek adına göz ardı edilmemelidir (RIBA ve ARUP, t.y).

RIBA ve ARUP, mimarlar ve kentsel tasarımcılar için veri ile çalışmanın dört ana yaklaşımını şu şekilde tanımlamaktadır:

1. Mimarların, kullanıcı ihtiyaçlarını tanımlamaya yardımcı olmak için verileri kullanması: İnsanların kamusal alanları ve altyapı sistemlerini nasıl kullandıkları ile ilgili güncel zamanlı veri türleri, kullanıcı ihtiyaçlarını daha iyi anlayarak bu ihtiyaçları daha iyi karşılayan yapılar, kentler oluşturmaya yardımcı olmaktadır.
2. Verileri kullanarak modelleme, analiz ve benzetim(simülasyon) yapılması: Veri ve modelleme araçları, tasarımcıların tasarımlarını yapım sürecine girmeden önce test ederek zamandan tasarruf etmelerini ve diğer tasarım opsiyonları üzerine düşünmelerini sağlamaktadır.
3. Yerel ve ulusal politika oluşturma ve uygulamasını iyileştirmek için verilerin analiz edilmesi: Kentlerin, vatandaşlara planlama ve hizmet sunumunu iyileştirmek için sahip oldukları çok miktarda veriyi, sorunları belirlemek ve çözmek için kullanma potansiyeli bulunmaktadır.
4. Mimari tasarım süreçlerini geliştirmek için verilerin kullanılması: Binalarda veriler, aşağıda belirtildiği gibi çok çeşitli kaynaklardan oluşmaktadır:
 - Tasarım ve inşaat (örneğin yapı bilgi modellemesi)
 - Kullanım sonrası değerlendirme
 - Kamu hizmetleri, inşaat hizmetleri, bina yönetim sistemleri vb.
 - Altyapı ve ulaştırma sistemleri
 - Satın alma sistemleri, performans raporlaması, iş planlaması ve benzeri kurumsal sistemler
 - Bakım ve değiştirme sistemleri
 - Operasyonel maliyet izlemesi
 - Bilişim teknolojisi sistemleri ve ekipmanları

Bu kaynaklardan gelen veriler performansı değerlendirmek, piyasa rekabet gücünü artırmak ve kaynakları tahsis etmek gibi nedenlerle kullanılabilir. Tarihsel olarak bu verileri toplamak zor ve pahalı olmuştur. Verilerin, yapı ve format çeşitliliği gibi nedenlerden dolayı, kâğıt kayıtlarından sayısal sistemlere manuel olarak aktarılması gerektiği için kullanımı zor olmuştur. Bu nedenle geçmişte veri uygulamalarının üst

düzyer karar verme yönteminde kullanılmasından ziyade belirli teknik işlevlerde sınırlı olarak kullanıldığı görülmektedir (RIBA ve ARUP, t.y).

Günümüzde, nesnelerin internetinin (internet of things) ortaya çıkışı, geliştirilmiş veri standartları, büyük veri analitik teknolojileri ve görselleştirme teknikleri, karar vericilerin karmaşık verileri çeşitli kaynaklardan anlamalarına ve sorgulamalarına olanak tanıyarak, bu sorunun üstesinden gelmeyi mümkün kılmaktadır. Akıllı teknolojiler, büyük miktarda verinin toplanmasını, depolanmasını, analiz edilmesini ve dağıtımını sağlamaktadır. Bu teknolojiler kullanılarak, binaların daha akıllı ve hızlı tepki vermesi ile birlikte performansları iyileştirilebilmektedir. Böylece otomatikleştirilmiş sistemler, akıllı bina yönetim sistemleri, enerji verimliliği önlemleri, kablosuz teknolojiler, dijital altyapılar, uyarlanabilir enerji sistemleri, ağıba bağlı cihazlar, veri toplama cihazları, bilgi ve iletişim ağı, yardımcı teknolojiler, uzaktan gözlemeleme, arıza teşhisi ve belirtileri gibi teknolojiler ile yenilikler sağlanabilmektedir.

2.5.3 YBM verisi

Yapı Bilgi Modellemesi, Uluslararası Yapı Bilgi Modelleme Standartları Komitesine (National BIM Standards Committee, 2007) göre “Bir tesisin fiziksel ve işlevsel özelliklerinin dijital bir gösterimi olarak tanımlanmaktadır. Bu nedenle YBM, binanın yaşam döngüsü boyunca alınacak olan kararlar için güvenilir bir bilgi kaynağı olarak hizmet etmektedir.” YBM, bina verilerini ve istatistikleri oluşturmak, değiştirmek ve yönetmek için bir veri tabanı, farklı proje üyeleri arasında iletişim kurmak ve bilgi paylaşmak için bir platform ve kapsamlı ve nesneye yönelik bina modelleri oluşturmak için bir araçtır.

Bilgisayar destekli tasarım (Computer Aided Design-CAD) ve Yapı Bilgi Modellemesi (YBM) arasındaki temel fark, sisteme veri eklenebilmesi yeteneğidir. YBM tabanlı projeye eklenen veriler model analizi, ürün seçimi, maliyet tahmini, şartname hazırlanması ve tesis yönetimi için kullanılabilir. Yapı bilgi modeli eklenebilen, manipüle edilebilen, görüntülenen, çıkarılan ve ihtiyaç duyulmadığı durumlarda silinebilecek verilerden oluşan bir dizi tablo içeren bir ilişkisel veri tabanına sahiptir. Modele eklenen her veri düzgün bir biçimde oluşturulduğunda bu veriler özel yazılımlar ile birlikte analiz için kullanılabilir veya tesis yönetimi için saklanabilir. Bu nedenle modele eklediğimiz veriler arttıkça bu verilerin proje yaşam

döngüsü boyunca kullanabilmesi için belirli standart ve formların oluşturulması gerekmektedir. Ayrıca sistem içerisinde oluşturulan mevcut bilginin yönetimi için de YBM tabanlı ek yazılımlara ihtiyaç duyulmaktadır.

Modelin geliştirilmesi, genellikle binanın sahip olduğu, olacağı bir dizi bileşenlerin (family) modele eklenmesiyle başlamaktadır. Bu birincil bileşenler muhtemelen projede en önemli ve en çok kullanılan bileşenlerdir. Proje başlangıç aşamasında bu bileşenler hakkında şartname ile ilgili herhangi bir bilgiye sahip olunmaması normaldir. Ancak projeye eklenen ürünün, bileşenin neye benzediğini grafiksel olarak ortaya koymakla birlikte ilerleyen süreçlerde performans değerlerinin ne olduğu gibi bilgiler modele işlenmektedir. Öznelikler bir modele eklendiğinde, hemen tüm değerlere ihtiyaç duyulmamaktadır. Örneğin, projede yer alan tüm kapıların yangına dayanım süresinin projenin başında modele eklenmesine gerek duyulmayabilir. Bu nedenle farklı proje aşamalarında hangi bilgilerin sisteme dâhil edileceğini ve katılımcılar arasında ne zaman ve nasıl bir bilgi paylaşımı olacağına karar verilmesi ve bu doğrultuda bileşenlerin parametrelerinin düzenlenmesi oldukça önemlidir.

Proje içerisinde tüm bileşenlere ait bilgiler sisteme eklendiğinde yapı bilgi modeli şartname oluşturmak için şartname hazırlayıcısına teslim edilebilir. Böylece veri değişimi nedeniyle ortaya çıkan tutarsızlıkların önüne geçilebilmektedir. Ayrıca, piyasada bu süreci daha da otomatikleştirmek için tasarlanmış yazılımlar mevcuttur.

Yapı bilgi modeline eklenen her bir özellik, en uygun ürünleri seçmek ve projenin tasarım kriterlerinin karşılayabildiği seçenekleri teyit etmek için bir kontrol aracı olarak kullanılmaktadır. Projenin erken tasarım aşamasında bileşenlerin performans değerlerinin, malzeme özelliklerinin ve bina yöneliminin birtakım analizler için kullanılması bu durumun bir göstergesidir.

Modele bilgi eklemek, tasarım sırasında analiz yapılabilmesi, ürün tedarikinin basitleştirilmesi ve tesis yönetimi için dijital bir kullanım kılavuzu oluşturulması gibi avantajlarla mimarlık, mühendislik ve inşaat sektörüne YBM sisteminin getirdiği birçok kolaylık bulunmaktadır. Bir modelin içerdiği bilgilerin yalnızca mimarlar için değil, proje ile ilgili tüm katılımcılar için de faydalı olduğu açıktır. Bu nedenle bilgi modeline verilerin eklenmesi ve yönetimi ile ilgili bilinmesi gereken en önemli nokta, modele eklenen niteliklerin tutarlı, erişilebilir ve belirli standartlara sahip olması ile birlikte doğru zamanda oluşturulması ve paylaşılması gerektiğidir.

2.5.3.1 YBM sisteminde veri kullanımı

Yapı bilgi modelinin verimliliği eklenen verilere bağlıdır. Ürünler ve projeler için geçerli birkaç bilgi türü vardır; Bazıları grafik, bazıları bilgilendirici niteliktedir. Bilgi eklemekten önce son kullanıcının ihtiyaçlarını ve projenin nihai hedefini belirlemek önemlidir. Modelde gerekenden fazla veri seti oluşturma hiçbir katılımcıya faydası olmamaktadır. Aksine önemli miktarda zaman kaybı oluşturmaktadır. Modele eklenen veriler müşteri gereksinimleri, devlet zorunlulukları, tasarım kararları gibi birçok farklı kaynaktan olabilmektedir. Yapı bilgi modelinde yer alan bilgiler, boyutsal bilgi, kimlik bilgisi, performans bilgisi, uygulama bilgisi, sürdürülebilirlik bilgisi yönetim ve bakım bilgisi ile şartname bilgisi olabilmektedir (Weygant, 2011). Modelde kullanılan bu bilgiler, kısaca aşağıda açıklanmıştır:

Boyutsal Bilgi: Boyut bilgisi, bir projede yaratılan grafik öğeleri ile ilişkili boyutları, şekilleri ve alanları oluşturur. Ayrıntılı bileşenler, nesnenin yapımı ile ilgili birçok boyuta sahip olabilir. Başlangıçta nesne oluştururken, nesnelerin kullanılan yazılım içerisinde nasıl görüntüleneceğini, organize edileceğini ve oluşturulan nesnenin parametrik olup olmaması gerekliliğini düşünmek gerekmektedir. Belirli bir bileşen için hangi boyutların gerekli olduğu düşünüldüğünde, bu bilgilerin projeye nasıl yükleneceğine, nesneye ait hangi bilgilerin değiştirilebilir ve sabit olacağına, nesnenin kendisinden çıkarılacak olan bilgi türlerinin neler olabileceğine ve nesnenin hangi analizler için kullanılabileceğine karar verilmesi gerekmektedir.

Kimlik Bilgisi: Kimlik bilgileri, bir ürünün ne olduğunu belirlemek için gerekli olmaktadır. Nesnelerin, ticari isim, model numarası gibi niteliklerle kategorilere ayrılması ve bu bilgilerin daha sonra kolayca alınabileceği birkaç nitelikten oluşması gerekmektedir. Bir bileşenin işlevselliğini artırmak için, özel bileşenin tasarımına, tanımına, kuruluma, satın alınmasına veya bakımına yardımcı olması için referansla ilgili bilgilere bir dizi bilgi eklenebilmektedir.

Performans Bilgisi: Performans bilgileri, endüstri standartlarına dayanan bir ürünü nitelendirmekte ve proje ekibine bir ürünün seçilme nedeninin belirtilmesine yardımcı olmaktadır. Çoğu ürün, eklenebilecek önemli miktarda performans bilgisine sahip olmaktadır. Bir bileşenin hangi yönlerinin bir bina bilgi modeli içerisinde ilgili olduğunu belirlemek ve bunları ilgili bilgi olarak ilişkilendirmek, model içindeki bilgileri düzenli tutmak ve bilgi aşırı yüklenmesini önlemek için gereklidir. Bir modele

bir ürünün birden fazla performans bilgisi eklenebilmektedir. Bazı performans bilgileri bilgilendirici olmalarının ötesinde analitik niteliktedir. Pek çok YBM yazılım platformunda, projenin enerji ve yapısal performansını analiz edebilecek uygulamalar ve eklentiler bulunmaktadır. Bu eklentiler ve uygulamalarda analiz süreci için kullanılacak olan bilgiler yapı bilgi modelinden alınmaktadır, bu nedenle binanın termal performansı ya da yapısal analizi ile ilgili bileşenlerin (duvar, çatı, pencere, taşıyıcı sistem vs.) performans değerlerinin doğru ve eksiksiz bir şekilde sisteme girilmesi gerekmektedir.

Uygulama Bilgisi: Proje yüklenicisinin modeli en son kullanıcı olarak aldığı durumlarda kurulum, imalat ya da uygulama bilgilerinin modele eklenmesi ve belirlenen işi tamamlamak için yükleniciler tarafından yerine getirilmesi gereken kriterlerin sağlanması gerekmektedir. Projedeki bileşenler inşaat sahalarındaki iş akışına göre düzenlenebilmektedir. Bir bileşenin kurulumundan sorumlu iş için geçerli olan nitelikleri eklemek, bunları aşamalı olarak sıralamak, projedeki işlerin inşaat aşamasındaki yönetimini kolaylaştırmaktadır. Ayrıca modelin ve bileşenlerin doğruluğuna bağlı olarak, modelin ürün imalatı için büyük bir potansiyeli bulunmaktadır. Bazı ürün tedarikçileri kendi YBM kütüphanelerini kullanıcılara sunmaktadır. Böylece modele, üretimi yapacak olan firma tarafından verilen bilgiler doğru bir şekilde eklenebilmektedir.

Sürdürülebilirlik Bilgisi: Sürdürülebilirlik kavramı, günümüzde mimarlık, mühendislik ve inşaat sektörünün gündeminde ön sıralarda yer almaktadır. Mimarlar, mühendisler mümkün olan en verimli binayı inşa etmek istemektedirler. Enerji etkin tasarımda enerji tasarrufuyla binanın ömrü boyunca tasarruf sağlamaya çalışılmaktadır. Sürdürülebilirlik ile ilgili belirli değerler, bileşenler aracılığıyla projeye eklendiğinde, bilgilerin derecelendirme sisteminin nasıl çalıştığını anlayan ve belirli bileşenlerin derecelendirmenin üzerinde ne kadar etkili olduğunun belirlenmesini sağlayan kullanıcılar tarafından çıkarılması ve analiz edilmesi gerekmektedir.

Modele sürdürülebilirlik bilgisinin eklenmesi, projeye ait bir bileşenin değiştirilmesi gereken sıklığı, ürünün binanın sakinleri için yaşam kalitesini nasıl etkilediğini ve binanın yaşam döngüsü boyunca bileşenlerin bakım maliyetini ortaya koyabilmektedir. Kullandıkları enerji miktarına veya binanın sürdürülebilirlik derecelendirmesinde ne kadar etkili olduklarına göre proje bileşenlerini seçmek kadar

binanın yönetimi ve işletmesi aşamasında bu ürünleri doğru kullanmak ve bakımını yapmak önemlidir. Bu nedenle modelde yer alan önemli miktarda bilginin, yaşam döngüsü değerlendirmesi, tesis bakımı ve bütçeleme için kullanılması amacıyla biçimlendirilmesi gerekmektedir.

Yönetim ve Bakım Bilgisi: Bir yapı bilgi modeli, tesis yöneticileri ve bina kullanıcıları için dijital bir kılavuz niteliği taşımaktadır. Model, yapılmış olan projenin aynısı olarak temsil edildiğinde ve bina kullanıcılarının kullanımına sunulduğunda, işletme yönetimi, bakım ve bütçeleme gibi konular için bu bilgiler kullanılabilir. Bina bilgi modeli, binaların HVAC, güvenlik ve yangın koruma kontrol sistemlerinin uzaktan yönetilmesine olanak tanıyan üç boyutlu bir kullanıcı arabirimi oluşturmak için kullanılabilir. Bir projede kullanılan her bileşen, binanın yaşam döngüsü boyunca bakım ve değişim gerektirmektedir. Yapı bilgi modeli, tesis yönetimi ile ilgili bileşenlerin rutin bakım ve değişim zamanlamasının öngörülmesini sağlamaktadır. Proje bileşenleri için bu tür bilgilere sahip olmak, binaya uygun bir şekilde bakılmasını ve bakım gerektiren elemanların yanlışlıkla gözden kaçırılmamasını sağlamaktadır. Bileşen veya sistemin incelenme sıklığını belirten bir gösterim eklenerek, tesis personelinin rutin bakım yapması gerektiğini gösteren zaman çizelgesi oluşturulabilir.

Gelecek için ise bir yapı bilgi modeli, farklı odalardaki sıcaklığın istenilen değerlere göre ayarlanmasına veya bir güvenlik sensörü uyarıldığında kameranın otomatik olarak etkinleştirilmesine izin verebilecek düzeye gelecektir. Gereken bilgiler, modele eklenildiğinde proje için birden çok otomasyon imkânı oluşmaktadır.

Şartname Bilgisi: Şartname bilgileri, projede hangi bileşenlerin kullanıldığını göstermekle birlikte bu bileşenlerin neden seçildiğini belirlemek için kullanılan veri veya nitelikler kümesi olarak da tanımlanabilmektedir.

Yapı bilgi modelinde veriler oluşturduktan sonra süreç içerisinde verileri kullanan farklı katılımcılar bulunmaktadır. Modeldeki verilerin kullanıcıları mimarlardan şartname hazırlayıcılarına, mühendislerden danışmanlara ve bazı durumlarda yüklenicilere kadar projenin tasarımıyla ilişkili olan kişiler olabilmektedir. Bu bireylerin her biri süreç içerisinde bilgiye katkı sağlarken aynı zamanda da gerekli bilgileri kendi amaçları doğrultusunda kullanmaktadırlar. Bu durum modeldeki verilerin ilgili disiplinler aracılığı ile artmasına izin vermektedir. Mimarlara bir bileşene

tasarım özellikleri ekleyebilir, şartnameyi hazırlayanlar bir bileşene ürün seçim bilgisi ekleyebilir, mühendisler bileşene performans ve kullanım bilgisi ekleyebilir ve danışmanlar uzmanlıkları dahilinde belirli nitelikleri modele ekleyebilmektedirler.

Projenin tasarımı sırasında, mimarlar YBM verilerini yapı analizi yapmak için kullanmaktadırlar. Enerji analizinden mekânsal ilişkilere, çeşitli bileşenlerdeki tasarım alternatiflerine ilişkin senaryolara kadar, tasarım sürecini hızlandırmak ve yapılandırmak için model içindeki veriler kullanılmaktadır. Mimarlar, istenilen estetik kaliteyi yakalamak için öğeleri boyutlarına, biçimlerine veya görünümüne göre değiştirebilmektedirler. Bu görsel temsil ve analizler, tasarımın nasıl ilerlediğini analizler ile birlikte ortaya koyarak kullanıcı ya da işveren tarafından tasarımın onaylanarak ilerleyebilmesini sağlamaktadır. Bir projeye yüzlerce öğe eklendiğinde, bileşenlerin birbirleriyle etkileşime girmesi kaçınılmazdır. Bu durum bazen istenilmeyen çakışmalara sebep olmaktadır. Mimarlar, bir kategorideki bir bileşenin, projenin herhangi bir yerinde başka bir kategorideki bir bileşene müdahale edip etmediğini gösteren kontrol çakışmalarının listesini elde edebilmektedir. Çakışma kontrolü şantiye sırasında karşılaşılabilmesi muhtemel birçok sorunun önceden fark edilip ilgili kişiler tarafından çözülebilmesini sağlamaktadır.

Şartname hazırlayıcısı inşaat belgelerinin oluşturulmasıyla ilgili sorumlulukların yanı sıra proje sürecinde veri geliştirme ve yönetiminde etkin bir rol oynayabilmektedir. Yapı bilgi modelindeki verilerin yönetimi, sektörde yaygın olarak kabul gören standartların ve formların sistem içerisinde ne kadar doğru uygulandığı ile yakından ilgilidir. Şartname hazırlayan kullanıcı grubu en büyük miktarda bilgiye sahiptir, bu durumda proje modeli içerisindeki bilgilerin düzenlenmesinde ve yönetilmesinde oldukça önemli sorumluluklara sahip olmaktadır. Modeldeki bilgilerin diğer kullanıcılar tarafından artırılması ve düzenlenmesi ile birlikte model üzerinden inşaat dokümanlarının oluşturulması kolaylaşmaktadır. Modelle ilişkili veriler düzenli bir şekilde yönetildiğinde projede hangi öğelerin kullanıldığı ve hangi şartname bölümüne ait oldukları bilgisi kullanıcılar tarafından kolayca ortaya koyulabilmektedir. Bu durum şartname bilgilerini ön planda tutarak bir proje el kitabı oluşturmak için gerekli süreci büyük ölçüde azaltmaktadır.

Mühendisler, binanın güvenli, sürdürülebilir ve olabildiğince verimli olabilmesi için gerekli olan yük ve kullanım hesaplamalarını YBM verilerinden yararlanarak yapabilmektedirler. Yapılan analizler, modelin içine uygun olarak yerleştirilmiş

bilgileri temel almakta ve olması gereken değerler projenin ihtiyaçlarına göre belirlenmektedir. Gerekli bilgileri modelin içine yerleştirmek hem bilgi hem de grafik bakımından analiz edilmesini sağlamaktadır. Mekanik ve yapısal mühendisler, disiplinleri hakkında yeterli miktarda bilgiye yapı bilgi modeli içerisinde sahip olduklarında tasarım sırasında analiz yapmaları ve iş akış şemalarını düzenlemeleri mümkün olmaktadır.

YBM verileri, yükleniciler için metraj hesapları yapmayı, çakışma kontrolü ile projenin tüm disiplinleri ile birlikte inşa edilebilir olduğunu onaylamayı, problemli alanlarda yeni çözüm önerileri geliştirerek hataları ortadan kaldırmayı sunmaktadır. Mimarlar modelin verilerini dışa aktardığında ya da doğrudan modeli yüklenicilerin kullanımına sunduğunda, yükleniciler modeldeki bilgileri maliyet tahmini, proje zamanlaması, şantiye programlaması gibi işler için kullanabilmektedir. Hem grafiklere hem de verilere erişebilen yüklenici grubu projenin şantiyede nasıl ilerleyeceğini zaman simülasyonu ile görsel olarak analiz ederek projeyi en kısa sürede tamamlayabilmektedir. Birbirinden farklı iş gruplarının sahaya girebilmesi için birbirlerini beklemek yerine, iş gruplarının tamamen tamamlanmadan saha içerisinde aynı anda çalışılacak şekilde planlanmasına izin verebilmektedir. Sahaya getirilmesi gereken malzemeler için uygun zamanlı teslimat planları yapılabilme ve malzemeler saha içerisinde uygun yerlere yerleştirilebilmektedir. Sahaya getirilen malzemelerin saha içerisinde birkaç kez farklı yerlere taşınmalarının önüne geçilebilmektedir. Böylece yükleniciler yapı bilgi modelini kullanarak maliyet tasarrufu yapmaktadırlar.

Projenin teslim aşamasında, tesis yöneticileri ve sahiplerine tesise ait “dijital el kitabı” verilmekte; bu sayede tesis yöneticilerine tesisin periyodik bakımın planlanması, tesiste meydana gelebilecek değişiklikler için bütçe tahminin yapılabilmesi ve tesisin kullanımının izlenebilmesi gibi konularda kolaylıklar sağlanmaktadır. Bu nedenle proje tasarım aşamasından teslim aşamasına kadar olan süreçte yapı bilgi modelindeki verilerin doğru bir şekilde sisteme dâhil edilmesi ve yönetilmesi tesis yönetimi için oldukça önemlidir.

2.5.3.2 YBM sisteminde veri yönetimi

Yapı bilgi modeli içerisinde yer alan verilerin, sisteme ne zaman ve nasıl dâhil edilmesi gerektiği, sistem içerisinde nasıl yönetileceği ve kullanıcıların verilere ihtiyacı

olduğunda bunları nasıl edinebileceği gibi konuların yapı bilgi sistemleri içerisinde tariflenmesi gerekmektedir.

Binaların erken tasarım aşamasından inşaat dokümantasyonuna kadar olan süreci, tesis yönetimine temel olarak geliştirildiği için bina verilerinin toplanması ve yönetimi süreç içerisinde oldukça önemlidir. Bu sürecin yönetilmesi için sektör katılımcıları birtakım yöntemler, yazılımlar geliştirmişler ve geliştirmeye devam etmektedirler.

Yapı bilgi modeli, projede kullanılan tüm objeler ve bileşenlerin yanı sıra projeyi bir bütün olarak barındıran bir veri tabanıdır. Model içerisinde bina verileri olarak sayılabilen duvar, kapı gibi bileşenler, mahaller, mobilyalar olabilirken proje verileri olarak da proje bütçesi, proje zaman çizelgesi, proje katılımcılarının kimler olduğu ve gerçekleştirilecek işler ile ilgili gerekli açıklamalar bulunabilmektedir. Projeye eklenecek olan elemanın başka bir projede kullanılma ihtimalinin olduğu düşünülerek gerekli biçimlendirmelerin yapılması gerekmektedir. Projeye eklenecek olan bileşenler için oluşturulan yapı bilgi modelinin dışında bileşenlerin bulunduğu bir kütüphane oluşturulması bileşenlerin yönetimi ve başka bir projede kullanılmak üzere hazır hale getirilmesi için faydalı olmaktadır.

Proje süreci içerisinde farklı veri kaynaklarından veriyi YBM yazılımlarına aktarabilmek gerekmektedir. Örneğin, yapı kullanıcısının ihtiyaç programı alındığında, bu veriyle çalışılabilmesi için verinin yapısal bir veri tabanına dönüştürülmesi ve daha sonra yapı bilgi modelinde ihtiyaç programı gereksinimlerini karşılayabilmesi için kullanılması gerekmektedir. Veri kaynağı olarak mimarın elinde pdf veya excel formatında tablolar olabilir. Başlangıç olarak çözülmesi beklenen mimari problemlerin çözümünde ilk olarak, verilerin anlaşılması ve kullanılması için uygun hedefler belirlemek önemlidir. Ardından, bu hedeflere ulaşmak için gerekli yazılım ve eklentilere başvurulmaktadır. Bir YBM yazılımı olan Revit'in kendi dışa aktarım özelliği ile modelden veri çıkışı sağlanmaktadır. Ancak dışarıya aktarılacak olan verinin büyüklüğüne bağlı olarak bu yaklaşım zaman kaybına neden olabilmektedir. Bu nedenle bu süreci otomatikleştirmek için hesaplamalı tasarım araçlarından yararlanılmaktadır.

Yapı bilgi modelinin kalite kontrolünün yapılması süreç içerisinde YBM içeriğinin geliştirilmesi ve korunmasının en önemli unsurlarından birisidir. Yapı bilgi modeli, inşaat çizimleri ve şartnameler oluşturmak için kullanılacağı için içerisinde yer alan

verilerin istenilen standartlara uygun şekilde sürecin sonuna kadar kontrollü bir şekilde oluşturulması ve korunması gerekmektedir. Modelin içeriğinin kontrollü bir şekilde iletilebilmesi sadece grafik gösterimi ile ilgili olmamakla birlikte modelde ürün ve proje ile ilgili veriler bulunduğundan dolayı oldukça önemlidir. Bir yapı bilgi modelinde yer alan verilerin açık, eksiksiz, doğru bir şekilde sisteme dâhil edilmesi ve ilgili diğer belgelerle tutarlı olması beklenmektedir. Yanlış veriler yanlış ürünün belirlenmesine ve üretim olarak da yanlış ürünün üretilmesine dönüşebilmektedir. YBM içeriğine veri eklendiğinde, bu verilerin doğru veriler olduğundan ve sisteme doğru bir şekilde uygulandığından emin olmak için verilerin dikkatle eklenmesi ve sürekli güncellenmesi beklenmektedir. Veri tabloları ve referans e-tabloları, genellikle daha kolay bir şekilde veri güncellemesine izin vermekte ve ayrıca üretici, tedarikçi ile tasarımcı arasında bir köprü oluşmasını sağlamaktadır. Farklı ortamlar ile YBM ortamı arasında veri paylaşımı yapabilmek ve süreci optimize edebilmek için hesaplamalı tasarım araçlarından yararlanılmaktadır.

YBM verileri ve grafiklerinin kontrolü YBM yazılımı içerisinde kullanıcı odaklı olarak yapılabilmektedir. Fakat bu oldukça zaman alan bir yaklaşım olduğu için verilerin kalite kontrolünün yapılması için hesaplamalı tasarım yöntemlerine başvurulmaktadır. Modelin kalite kontrolünün sağlanması için hesaplamalı tasarım ortamında algoritmalar hazırlanmaktadır. Modele bilgi ekleyen katılımcıların bu algoritmalar vasıtası ile kendi ekledikleri bilgilerin doğruluğunu test edilebilmesiyle zaman ve veri yönetimi açısından kolaylıklar sağlanmaktadır.

Kalite kontrolü için bir prosedür oluşturmak, içerik geliştirme sürecinin düzene koyulmasını ve gereksiz iş yükünün en aza indirilmesine olanak tanımaktadır. Kalite kontrol sürecine modüler bir yaklaşım getirmek, modeli test etme sürecinin azalmasını sağlamaktadır. Her şeyden önce YBM sistemi içerisinde farklı disiplinlerle birlikte çalışabilmek ve projeyi iletilebilmek için modeldeki içeriğin, grafiklerin ve verilerin doğru ve eksiksiz olması gerekmektedir. Kalite kontrol sürecinin modeldeki grafik, veri, montaj bilgisi ve uygulamasına ilişkin bilgilerin doğruluğunun test edilmesi gerekmektedir.

Grafikler üzerinde kalite kontrolü gerçekleştirmek muhtemelen en zor olan ve en çok göz ardı edilen kısım olmaktadır. Oluşturulan bileşen içindeki grafikler, genel olarak, nesnenin tamamını oluşturmak için birlikte çalışan birkaç katıdan oluşturulmaktadır. Geliştirme sırasında nesnenin parametrelerinin çalışıp çalışmaması ile ilgili yapılan

kalite kontrolünün, proje boyunca test edilmesi gerekmektedir. Çoğu durumda, bazı katılar birbirlerine bağlı olacak şekilde modellenebilmektedir (nested family). Fakat bu yöntem tercih edildiğinde iç içe geçmiş bileşenler oluşturulacak olan çizelgelerde görünmemektedir. Bu nedenle bileşenlere ait hangi özelliklerin elde edilmesinin gerektiğine karar verilmesi ve buna göre modelleme yöntemi tercih edilmesi gerekmektedir. Metraj olarak elde edilmesi istenen malzeme bilgileri varsa bu bilgilerin de modelleme sırasında objelere eklenmesi gerekmektedir. Modelleme yöntemine karar vermek nihai hedeflerimizi belirlememizle ve hangi verilerin ihraç edileceği ile yakından ilgilidir.

YBM sistemini standart 2D, 3D grafiklerinden ayıran kısım verilerdir. YBM, bileşenleri, ürünleri, materyalleri, montaj bilgisini ve projeyi bir bütün olarak etkin bir şekilde depolayarak yönetebilmektedir. Oluşturulan veriler projedeki farklı noktalardaki farklı kişiler tarafından kullanılabilir. Bu nedenle projenin başlangıcından itibaren sistem içerisindeki verilerin doğru olması kritik önem taşımaktadır. Bir üreticiye özgü bileşen oluşturulduğunda ve bir projeye eklendiğinde içerdiği tüm bilgilerin doğru olması gerekmektedir. Bu nedenle, modelde yer alması planlanan bilgilerin kalite kontrolü her şeyden önce gelmektedir. Niteliklerin doğru bir şekilde atanmasına ve istenmeyen sonuçlar elde edilmeyeceğine emin olmak önemlidir. Bir projedeki pek çok bileşenin yüzlerce farklı malzeme, renk ve boyut gibi parametreleri vardır. Yapı bilgi modelinde artan parametrelere bağlı olarak zorlaşan veri yönetimi için mimarlık, mühendislik ve inşaat sektöründe yer alan disiplinler, hesaplamalı tasarım araçlarından faydalanmaktadır. Bir yapı modeli oluştururken geometrik ve geometrik olmayan verilerin kolayca ve hızlı bir şekilde güncellenebileceği bir yapı geliştirmek, uzun vadede modeli yeniden geliştirme konusunda zaman ve maliyet tasarrufu sağlamaktadır.

Yapı bilgi modellemesinin sektörde kullanılmasıyla birlikte, proje tasarımı sırasında bilgi ve işbirliğinde önemli miktarda artış olmuştur. Pek çok farklı disiplin arasında çok fazla bilgi alışverişi olduğundan, planlar, şartnameler, model ve diğer destekleyici belgeler arasında tutarlılık sağlamak için bu bilginin yönetilmesi oldukça önemlidir.

Şematik tasarım, proje başlangıç belgelerinden başlayarak yapının bileşenlerine daha spesifik özellikler ekleyerek yapı bilgi modelini geliştirmektedir. Tasarım geliştirme aşamasında, proje bir bütün olarak bileşenlerle birlikte geliştirilmektedir. Teklif vermeden önce, bu bilgiler, tedarikçi ve yüklenicilerin proje üzerinde kullanılacak

gerçek ürünleri belirlediği temel gereklilikler olarak hareket etmektedir. Bilgileri bileşenle birlikte modele eklemek, yapının belgelendirilmesi aşaması sırasında özniteliklerin modelden çıkartılmasını sağlamakta, böylece bir dizi çizelge oluşturulabilmektedir. İnşaat evresi, şematik tasarım ve tasarım geliştirme aşamasında elde edilen tüm bilgileri almakta ve bir projenin teklif aşamasında kullanılan çizimlerde bulunan bilgileri tamamlayan bir belge haline getirmektedir. Bir projenin inşası tamamlandıktan sonra, bir modelin içerdiği bilgiler doğru olduğu sürece, işletme sahibi ve tesis yöneticisi için önemli değer taşımaktadır. Yapı bilgi modeli proje sahibinin tüm projenin dokümantasyonunu başından sonuna kadar sağlamakta ve tesisin dijital el kitabı olarak kullanılabilmektedir. Tesis yöneticileri, yapı bilgi modelini bileşenlerin bakım ve yenilenmesini tahmin etmek için ve gelecekteki tesis geliştirilmesinde bütçe oluşturulması için kullanabilmektedir. Proje bilgileri, projedeki tek bir kişinin sorumluluğundan fazlasını gerektirmekte ve tasarımdan önce yapım sonrasına kadar uzanmaktadır. Bu nedenle verilerin katılımcılar tarafından denetiminin yapılması oldukça önemlidir. Bu bağlamda, ofislerde verinin yönetilmesi için yeni yaklaşımlar oluşturulmakta ve birtakım yardımcı yazılımlardan yararlanılmaktadır. Verilerin yönetilmesi için ele alınan YBM tabanlı iş akış yöntemleri ilerleyen bölümlerde detaylı olarak incelenmektedir.

3. HESAPLAMALI TASARIM

Hesaplamalı tasarım, yapının sanal bir modelinin doğrudan kullanıcı tarafından oluşturulması yerine tanımlayıcı bir betiğin çalıştırılmasıyla oluşturulması yöntemidir. Hesaplamalı tasarımın daha kapsayıcı tanımı ise sürecin bilgisayardan yardım alarak yönetilmesinin yerine, sürecin tamamının bilgisayar ortamında gerçekleştirilmesidir.

Humppi ve Österlund, Algoritma ve hesaplamalı tasarım için kullanılan ortamın “Algoritma Yardımlı Modelleme” (İng.: Algoritm Aided Design-AAD), “Parametrik Tasarım” veya “Hesaplamalı Tasarım” olarak adlandırıldığını ve bu ortam için kesin bir tanım olmadığını ifade etmiştir (Humppi ve Österlund, 2011).

Hesaplamalı tasarım yöntemi olarak metin kodu oluşturmak programlamanın bir yoludur ve dil özel sözdizimini anlamak mimarlar tarafından zor olmaktadır. Ayrıca programcı olmayan kullanıcılar için daha uygun olan bir programlama dili mevcuttur: görsel programlama. Bu bölümde görsel programlama detaylı bir şekilde ele alınmaktadır.

Bu bölümde, mevcut literatür incelemesi ile birlikte aşağıda bahsedilen konular araştırılmaktadır. Mevcut yapı bilgi modelleme sisteminde veri oluşturulması, kullanımı ile ilgili sınırlamalar ve bunları aşma yolları bölüm 3.1’de, görsel programlamanın tarihçesi bölüm 3.2’de, görsel programlamanın ilkeleri bölüm 3.3’de ve hesaplamalı tasarımdan YBM’ye doğru başlığı bölüm 3.4’de ele alınarak bölüm 3.5’de değerlendirme yapılmıştır.

3.1 Mevcut YBM Sisteminde Veri ile İlgili Sınırlamalar

Yapı bilgi modelleme ortamlarında üretilen dijital modellerin tipik özellikleri, veri açısından zengin, nesne yönelimli, akıllı, parametrik, bilgi paylaşımı açısından uygun olması olarak tanımlanmaktadır. Buna ek olarak kullanıcı bilgi yerleştirme, çıkarma, güncelleme ve değiştirme yeteneğine sahiptir. Günümüzde MMİ sektöründeki bilginin ve geri bildirimin önemli bir kısmı kullanılan yazılım tarafından üretilmektedir. Yazılım, bir binanın performansını hesaplama ve hatta binanın imar yasalarına uyup

uymadığını belirleme yeteneğine sahiptir (Braumann ve Brell-Cokcan, 2014). Ancak, mevcut yapı bilgi modelleme yazılım araçları, proje sürecinde yaygın bir şekilde veri saklama alanı olarak kullanılmasının yanı sıra tasarım süreci için yazılım olanaklarından yeterince yararlanılmamaktadır. Yazılımın tasarım süreci içerisinde kullanıldığı durumlarda da yazılım ile manuel modelleme yaklaşımı söz konusudur. Burada asıl sorun, tasarımcıların bilgisayarın hesaplama gücünden tamamen faydalanamamasıdır (Terzidis, 2006).

Coenders, proje süreçlerinde kullanılan mevcut yazılım uygulamalarının basit olduğunu ve YBM vizyonundan uzak olduğunu savunmaktadır. Mimari tasarım sürecinde uygun hesaplamalı tasarım yaklaşımlarının kullanılmaması ve yazılımlar arası birlikte çalışabilirliğin olmaması bilgi kaybına yol açmaktadır (Coenders, 2009). Hesaplamalı tasarım yöntemlerini kullanabilmek için ortaya konulmuş birçok yaklaşım mevcuttur, ancak bu yaklaşımlar doğru bir şekilde kullanılmamaktadır. Genellikle bir problem çözme yöntemi olarak kullanılan prototipleme, verilere dayalı algoritmalar kullanmak yerine kullanıcının deneme yanılma yolu ile oluşturulmaktadır (Braumann ve Brell-Cokcan, 2014).

YBM yazılım araçlarının standart yetenekleri dışındaki ana motivasyonu açıklamak kolaydır: Çoğu standart YBM araçları başlangıçta ne veri yönetimi ortamları ne de işbirliği platformları olarak düşünülmüştür. Tarihsel olarak, YBM araçları nesne odaklı modelleme ve sanal montaj için kullanılmaktaydı. Gelişimleri boyunca, YBM araçları veriler ile ilgili fonksiyonları barındırmaya başlamışlardır. Bazı durumlarda, yazılım geliştiricileri standart YBM geliştirme platformlarına ek araçlar sunmaktadır. Modelleme ve veri yönetimi süreçleri arasındaki bağları güçlendiren uygulamaların kullanımında bir artış yaşanması muhtemel görünmektedir. (Holzer,2016).

Hesaplama(computation) ve bilgisayarlaştırma-bilgisayara aktarma (computerisation) arasındaki ayrımı anlamak önemlidir (Menges ve Ahlquist, 2011). Hesaplama, kural tabanlı anlayış, simülasyonlar (benzetim) ve yinelemelerle ilgilidir. Hesaplama, akılcılaştırma, akıl yürütme, mantık, algoritma ve tahmin yürütme ile insan zekâsını genişletmeyi amaçlamaktadır. Buna karşılık, bilgisayarlaştırma tasarımcının zihninde zaten kavramsallaştırılmış, önceden belirlenmiş ve iyi tanımlanmış nesnelerin ve süreçlerin sayısallaştırılmasını içermektedir (Terzidis, 2006).

Bilgisayara aktarma işleminde, yapı bilgi modellemesi yazılımları ile verilerin üretilmesi, değiştirilmesi, düzenlenmesi ve depolanması çoğunlukla tasarımcı tarafından manuel olarak yapılmaktadır. Verilere akıllı bir şekilde erişmek ve işlemek için yeni araçların ve yöntemlerin oluşturulmasına ihtiyaç duyulmaktadır. Çünkü mevcut YBM yazılım araçları esas olarak verileri saklama işine odaklanmakta ve olgun bir tasarım düzeyinde tasarım değişiklikleri ile başa çıkılmasında zorlanmaktadır (Coenders, 2009).

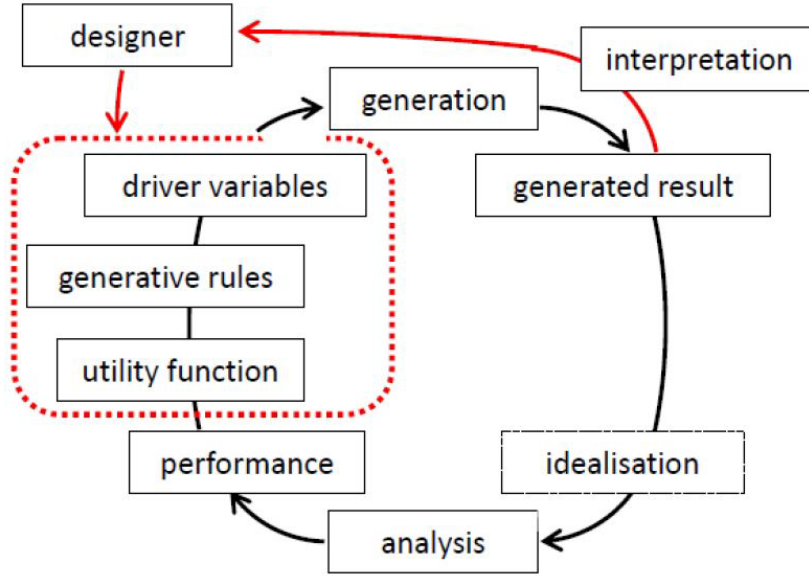
Proje süreci içerisinde yapı bilgi modelinde artan bilgi miktarı ve tasarımın hızlı döngüleri, mimarların ve mühendislerin denetimini artırmak için bir stratejiye ihtiyaç duymalarına sebep olmaktadır. Süreç içerisinde kurallar oluşturmak ve işlemek, optimize edilmiş tasarım çözümleri oluşturmak veya mevcut tasarım koşullarını standartlara göre doğrulamak bu stratejilerden sayılabilmektedir. Stratejiler uygulanırken kullanıcıların gerçek zamanlı geribildirim ile yapıya dair kararları hızlıca alabilmesi mümkün olmaktadır (Coenders, 2009).

Hedef, üretken mekanizmaları ve analitik / değerlendirici mekanizmaları birbirine bağlı tekrarlayıcı bir süreç içerisinde birleştirmek olmalıdır (Aish ve diğ., 2012). Olası kombinasyonların çokluğu bu süreci otomatik hale getiren bir yöntem gerektirmektedir (Şekil 3.1). Bu durum, kombinasyonları birbirine karşı değerlendirerek ve otomatik olarak yeni yapılandırmalar üreten mekanizmalar tarafından sağlanmaktadır (Aish ve diğ., 2012). Hesaplamalı tasarım, projenin geliştirilmesi sırasında tasarım parametrelerini, modelin analitik olarak değerlendirilmesine dayandırarak, tasarım sürecinin doğasının değiştirilmesine izin vermektedir (Kensek ve Noble, 2014).

Bir yapının kendinden beklenen performansı etkin bir şekilde karşılayabilmesi için potansiyel olarak birbiriyle çelişen birden fazla performans kriterinin dikkate alınması gerekmektedir (Aish ve diğ., 2012). Ayarlama ve yineleme ile hesaplama yoluyla çözülebilen karmaşık çok değişkenli denklemler, tasarımcının çok çeşitli potansiyel seçenekleri görmesini, anlamasını ve seçmesini sağlamaktadır (Kensek ve Noble, 2014).

İnsan, olası birçok seçeneğin ölçüt ve parametrelerinin gözden geçirilmesi ile birlikte seçeneklerin karşılaştırılması konusunda sınırlı yeteneklere sahiptir. Birçok özel gereksinimi olan yapılar için ileri karar teknikleri olmadan tasarımı optimize etmek

oldukça zordur. Elle kullanılan algoritma teknikleri sınırlıdır. Bu nedenle algoritmaları kullanan yazılımlarla tüm kriterleri ekleyerek yapılan optimizasyonlar önemlidir.



Şekil 3.1 : Proje tasarım optimizasyonu (Aish ve diğ., 2012)

Hesaplamalı tasarım ile birlikte optimizasyonlar gerçekleştirmenin temelinde algoritmalar bulunmaktadır. Algoritma teriminin çeşitli tanımları vardır ve bunlar aşağıda listelenmiştir:

“Bir algoritma, tam, eksiksiz ve genel bir şekilde açıklanan, basit, temel talimatların sonlu bir dizisidir” (Menges ve Ahlquist, 2011).

“Bir algoritma, bir problemi sonlu sayıda adımda mantıksal if-then-else işlemleri kullanarak önermek için kullanılan bir prosedürdür” (Terzidis, 2006).

“Bir algoritma, sonlu sayıda adımda açıkça tanımlanmış bir amaca hizmet etmeyi amaçlayan sınırlı bir talimat setidir. Bir algoritma bir değeri veya bir dizi değerleri girdi olarak alır, girdiyi dönüştüren bir dizi hesaplama adımını uygular ve son olarak çıktı olarak bir değer veya bir dizi değer üretir” (Dino, 2012).

“Bir algoritma, bazı değerler veya değerler kümesini girdi olarak alıp, çıktı olarak bazı değerler veya değerler dizisi üreten iyi tanımlanmış herhangi bir hesaplama prosedürüdür. Dolayısıyla bir algoritma, girdiyi çıktıya dönüştüren bir hesaplama aşamaları dizisidir. Bir algoritma, iyi tanımlanmış bir hesaplama probleminin çözülebilmesi için bir araç olarak da görülebilir. Sorunun ifadesi genel olarak istenen girdi / çıktı ilişkisini belirtir. Algoritma, bu girdi / çıktı ilişkisine ulaşmak için özel bir hesaplama prosedürünü açıklamaktadır” (Cormen, 2009).

Ayrıca Dino, algoritma tabanlı üretken bir işlemin 4 unsurunu belirtmiştir (Dino, 2012):

1. Başlatma koşulları ve parametreleri (giriş)
2. Üretken mekanizmalar (kurallar, algoritmalar vb.)
3. Opsiyonların üretilmesi hareketi (çıktı)
4. Yinelemeyi durdurma kararı ve en iyi opsiyonun seçilmesi

Algoritma, sıralama ve arama, veri yapısı işlemleri, kombinasyonel problemler, sayısal problemler ve hesaplama geometrisi gibi hesaplama problemlerini çözme yeteneğine sahiptir. Ancak nihai sonucun kalitesi, tasarımcının kodlama, yorumlama ve değiştirme becerilerine bağlıdır. Günümüzün güçlü bilgisayarlarıyla bile, zayıf tanımlı bir algoritma, zayıf veya hatalı sonuçlar üretmektedir. Güçlü bilgisayarların desteğiyle iyi tanımlanmış bir algoritma sorunu hızlı, verimli bir şekilde çözmekte ve doğru cevabı üretebilmektedir (Cormen, 2009).

Bir algoritma oluşturmak için, eylemlerin bilgisayar tarafından okunabilir bir programlama dili ile tanımlanması ve çalıştırılabilir bir programa derlenmesi gerekmektedir. Parametrik algoritmaların dijital simülasyon araçlarından elde edilen analitik sonuçlar ile birlikte kullanılması tasarımcının gücünü önemli ölçüde genişletmektedir. Böylece, optimize edilmiş ve etkili bir çözüm bulmak için birçok varyasyonu test etmek mümkün olmaktadır. Tasarımcı, tasarımın özelliklerini kontrol eden ve işleyen bir senaryo oluşturarak, tasarım amacının özelliklerini çözümün kendisine işlemektedir (Kensek ve Noble, 2014).

Hesaplamalı tasarım yöntemi olarak metin kodu oluşturmak programlamanın bir yoludur ve dil özel sözdizimini anlamak mimarlar tarafından zor olmaktadır. Programlamanın olanaklarından faydalanmak için zor olsa da tasarımcılar yeni bir bilgi alanı ile tanışmak zorunda kalmışlardır. Metinsel programlama araçlarına örnek olarak yapı bilgi modelleme yazılımı olan Revit için Makro Yöneticisi (Macro Manager) veya Revit Python Shell uzantısı verilebilir. Her ikisi de Revit yazılımında görevleri otomatikleştirmek için kullanılan metinsel programlama arabirimleridir. Ayrıca programcı olmayan kullanıcılar için daha uygun olan bir programlama dili mevcuttur: görsel programlama. Görsel programlama dilinin kullanıldığı araçlara örnek olarak Revit ile birlikte kullanılan Dynamo verilebilir. Mevcut YBM araçlarında yer alan tanımlı araçlarla temsil oluşturulmasının getirdiği kısıtlamalara karşı görsel

programlama aracı olan Dynamo ile birlikte kullanıcılar kendi araçlarını ve modelleme biçimlerini belirleyebilmektedirler.

3.2 Görsel Programlamanın Tarihçesi

En eski görsel programlama dillerinden biri, 1966'da William Sutherland tarafından tasarlanan Görsel Programlama Editörü'dür. 1963 yılında 'Sketchpad' yazan kardeşi Ivan Sutherland'den etkilenmiştir. Sketchpad, insan-bilgisayar etkileşimini ve Smith'in bilgisayar programında Pygmalion'u, grafik imajların bir programlama dili tasvir edebileceğini daha net hale getirmiştir (Blackwell ve Rodden, 2003). 70'lerin sonlarında akıllı tasarım sistemleri CAD ortamında kullanılabilir hale getirilmiş, ancak bu tür sistemlerin uygulamaya özgü oluşu ve genelliği sınırlı kalmıştır. Genel amaçlı görsel programlama dillerinin geliştirilmesi 80'lerin ortalarından beri gerçekleştirilmiştir (Hirakawa ve Ichikawa, 1992).

İlk girişimler tek boyutlu ve statik temsile sahip olan metinden tamamen kaçınmayı hedeflemiştir. Ayrıca görsel bileşenlerin önemli program parçalarını açıklamak için anlamlı bir etiket ve notlara sahip olması gerektiği ve metinsel programlamadan tamamen kaçınmanın pratik olmadığı iddia edilmiştir. Görsel programlama, programlama sürecinde anlamlı grafik temsillerin kullanılması olarak tanımlanmaktadır. Hirakawa ve Ichikawa (1992), görsel programlamayı görsel öğelerin manipüle edilmesiyle, yani kendiliğinden görsel olan bir dil aracılığıyla işlemler yapma eylemi olarak tanımlamaktadır.

Proje sürecinde karşılaşılan manüel görevleri en aza indirmek ve iş akışlarını, programcılara bağımlı olmaksızın otomatikleştirmek görsel programlama araçları ile mümkün olmaktadır. Mimarlar tarafından yaygın olarak kullanılan görsel programlama araçlarına örnek olarak Grasshopper ve Dynamo verilebilmektedir. Grasshopper, MMİ endüstrisinde iki boyutlu (2D) ve üç boyutlu (3D) geometrilerin manipüle edilmesinde uzun süredir kullanılan görsel programlama aracıdır. YBM araçları süreç içerisinde veri bakımından zengin dijital modeller kullanılarak verinin yönetilebilmesi için geliştirilmiştir. Sistem içerisindeki verileri otomatik olarak kullanmak için ortalama bir tasarımcının metinsel bir algoritma arabirimi ile çalışması sektördeki çoğunluk tarafından kabul edilmemiştir. Bu nedenle YBM yazılım araçlarına uygun görsel programlama araçları geliştirilmiştir. Bu amaç doğrultusunda Revit ve Navisworks için Dynamo aracı oluşturulmuştur.

Görsel programlamanın gelişimsel geçmişi, görsel programlamanın bugüne kadar sadece profesyonel yazılımcıların yazılım bileşenlerinin ve bunların birbirleriyle olan ilişkilerinin şematik tasarımı için kabul gördüğünü göstermektedir. Leitão ve Santos (2011), tasarım probleminin karmaşıklığı arttığında metinsel programlama dillerinin görsel programlama dillerine göre daha üretken olabildiğini belirtmiştir. Açıklamaları, belirli karmaşık senaryolar için kabul edilebilir; ancak son yıllarda MMİ sektöründeki mimar ve mühendislerin algoritmalarla tasarım problemlerini çözmek için ilave görsel programlama arabirimlerine olan talepleri giderek artmaktadır.

3.3 Görsel Programlama İlkeleri

Görsel programlama, metinsel programlama kavramını kullanmaktadır, ancak metinsel kod parçacıklarını görsel bileşenlerle değiştirerek soyutlamayı basitleştirmektedir. Görsel programlama dilleri de iyi tanımlanmış sözdizimi ve semantiklere sahiptir, ancak programcı olmayan kullanıcılar için daha uygun ve daha az soyut bir forma sahiptirler.

Görsel kod bloklarında sözdizimi hataları yoktur ve görsel olarak kullanıcıya düğümler şeklinde sunulur. Düğümlerin girişleri, çıkışları, giriş alanları, kaydırıcıları, düğmeleri vb. vardır. Düğmeler, kaydırıcılar, anahtarlar ve görsel ön izleme birimleri(watch) gibi görsel ara yüzler kullanıcılar tarafından oluşturulmaktadır. Görsel programlama dilinin oluşturduğu görsel etki, sezgisel kullanım, düşük hata riski, doğrudan geribildirim ve kolay manipülasyon yeteneği, kullanıcının motivasyonunu artırmaktadır.

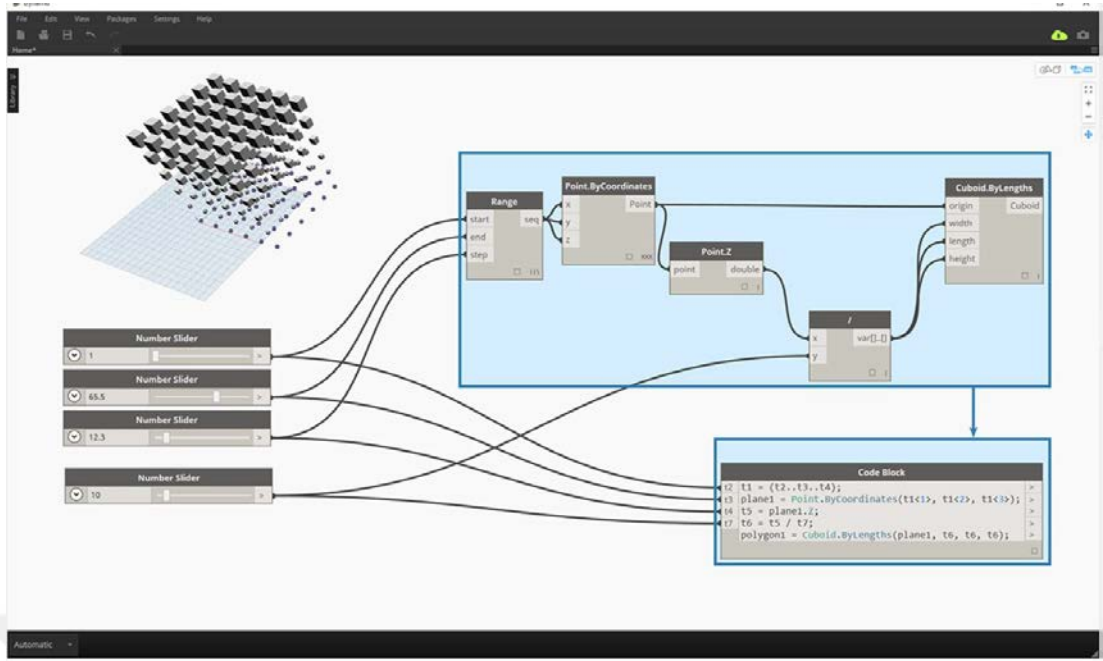
Gömülü ve önden test edilmiş kodlu görsel bileşenlerin kullanımı, metinsel programlama kod yazımında sıkça görülen yanlış kod yazımını engellemektedir. Görsel programlamada yer alan görsel bileşenlerin arkasında metinsel kod gizlidir. Bu esnek yaklaşımla, algoritmalar oluşturmak için soyut bir sözdizimini ezberlemek artık gerekli değildir. Böylece görsel algoritmalarındaki hataların başlıca kaynakları kod yanlış yazımından kaynaklı hatalardan mantıksal hatalara doğru evrilmiştir. Bu durumda, işlem zincirindeki hatalar tespit edilirse sistem, kullanıcının gereksinimlerine göre çok hızlı ve basit bir şekilde ayarlanabilmektedir. İşlem zincirindeki hataların daha kolay tespit edilebilmesi için de görsel programlama kullanımı ile ilgili birtakım yöntemler geliştirilebilmektedir. Düğümlerin yarı-soyut görünümü, düğmeler, kaydırıcılar, matematiksel ifadeler, simgeler ve düğümün

durumunu renklerle ifade eden grafik yardımcılar, bir algoritmanın mantığını anlamaya yardımcı olmaktadır. Düğümlerin üzerine yazılabilen etiketler ile de düğümlerin yorumlanmasına yardımcı olunmaktadır.

Gelişmiş kullanıcılar için görsel düğümleri metinsel kodlama dilleriyle genişletmek, bir metin düzenleyici arabirimine sahip özel kod bloğu düğümleri ile mümkündür. Komut dizgesi ile görsel programlama dili arasında geçiş yapabilme, birbirine haritalama yapabilme mimari tasarım eğitiminde son derece önemli olan soyut kavramları anlama kapasitesini geliştirmeye yardımcı olmaktadır (Celani ve Vaz, 2012).

DesignScript; Autodesk bünyesinde Robert Aish tarafından, tasarım ölçütlerini temel olarak geliştirilmiş olan, sadece modellemeye yönelik bir araç olmanın ötesinde bir programlama dili ve eğitsel bir araçtır. Bu dilin grafik programlama dillerindeki kısıtlamalarının azaltılması hedeflenmiştir (Aish, 2012). Görsel programlama araçlarının bileşenlerinin birbirine bağlanarak çalışması prensibinin yanı sıra, daha detaylı ve ileri düzey algoritmalar için “node-to-code” özelliği kullanılabilmektedir (Şekil 3.2). Bu özellik hazırlanmış olan görsel algoritmayı, metin tabanlı algoritmaya çevirir ki bu da kullanıcı açısından dilin eğitici yönünü vurgulamaktadır. Böylece kullanıcı programlama dilleri ile ilgili yeteneklerini geliştirebilmektedir. Bu özellik kullanılarak hazırlanan algoritmanın metin ve görsel yorumları oluşturulmaktadır. DesignScript ve Dynamo ilk başlarda birbirinden bağımsız iki farklı ortamken, geliştirilme sürecinde DesignScript’in bütün özellikleri ve geometri motoru Dynamo içerisine gömülmüştür ve hali hazırda da bu şekilde devam etmektedir.

Son on yılda, görsel programlama araçlarının işlevselliğini genişletmek için birincil programlama ortamına eklentiler kurmak yaygınlaşmıştır. Dynamo’da bulunan Paket Yöneticisi bu duruma örnek olarak verilebilmektedir. Eklentiler aracılığı ile oluşturulan düğümler ile mevcut uygulamada yer alan düğümler birbiriyle iletişim halindedir. İki yönlü bağlantılar oluşturmak için API (Application Programming Interface) komutları kullanılabılır. Örneğin, bir YBM yazılım aracından veri çıkartılabilir ve uygun düğümlerin yardımıyla Microsoft Excel ile çift yönlü bir bağlantı kurulabilir. Excel’deki verilerin manipüle edilmesinden sonra, veriler YBM yazılım aracına geri aktarılabilir. Böylece bilgi kaybı önlenir ve çözümleri optimize etmek için algoritmalar kullanılabılır. Bu konu ilerleyen bölümlerde daha detaylı bir şekilde ele alınmaktadır.



Şekil 3.2 : DesignScript NodeToCode özelliği ile algoritmanın dönüştürülmesi

Görsel programlamanın avantaj ve dezavantajları kısaca aşağıda belirtilmiştir.

Avantajları:

- Büyük motivasyon ve öğrenme potansiyeli barındırıyor olması
- Sözdizimsel yapıların azaltılması ve şematik ilişkilere vurgu yapılması
- Kullanıcı tarafından yapılan deneme yanılma işlemlerinin daha hızlı yapılabilmesi
- Görsel programlama araçlarında giriş parametresi olarak kullanılan kaydırıcıya (slider) ait parametreleri değiştirerek gerçek zamanlı bir şekilde tasarım sonuçlarının gözlemlenebilmesi
- Mimar ve mühendislerin görsel programlama araçları ile birlikte kendi problemlerine özel sanal simülasyon ortamları oluşturabilmeleri
- Tasarımcının karar vermesini destekleyen sürekli geri besleme sürecinin olması
- Grasshopper veya Dynamo gibi görsel programlama ortamlarında yer alan tüm temel işlevler programlamaya erişilebilir bir yaklaşım ile sunulduğu için kullanıcıların yeniden bir ara yüz geliştirmemeleri
- Ara yüz geliştirmek yerine kullanıcının, gerekli ek işlevlerin geliştirilmesine odaklanabilmesi

- Metinsel programlamaya göre yerine getirilmesi gereken daha az sözdizimsel komutlara ihtiyacın olması nedeniyle kullanıcıların algoritmayı oluştururken daha kolay bir şekilde ilerleyebilmesi

Dezavantajları:

- Eksik standartlara sahip olması
- Oluşan illüstrasyon yoğunluğu
- Metinsel programlama koduyla karşılaştırıldığında, karmaşık bir grafiksel algoritmanın aynı miktarda üretilen bilgi için çok daha fazla alan gerektirmesi

3.4 Hesaplamalı Tasarımdan YBM'ye doğru

YBM yazılımları, sınırlı akıllı davranışlara sahip olan ve oluşturulan parametrelerin değerlerini ayarlayarak yapılandırılan parametrik nesneler ile karakterize edilmektedir. Tasarımcı, manuel nesneleri oluşturur, yerleştirir, değiştirir ve sayısal bilgileri nesnelere ekler. Bu aşamada çoğunlukla eksik olan şey, proje düzeyindeki genel kontrollerin olmamasıdır (Boeykens, 2012). YBM yazılımları maliyet tahmini, yapısal hesaplamalar ve enerji analizi üretebilmektedir. Bu çalışmalar YBM tabanlı işlemin daha gelişmiş uygulamalarının mümkün olabileceğinin, modelin analiz ve simülasyon potansiyelinin olduğunun ve geliştirilebileceğinin erken göstergeleri olmaktadır (Kensek ve Noble, 2014).

Görsel programlama araçları, modelleme platformlarında daha birleştirici bir işbirliği türü oluşturabilmektedir. Günümüzde yapı bilgi modellerinde üretilen muazzam miktarda veriyle, bu verilere erişmek ve yönetmek, görsel programlama dillerini kullanmayanlar için zor olmaya devam etmektedir. YBM bağlamında, görsel bir programlama senaryosu, hesaplama algoritmaları ile yeni bilgiler üretmek için bir modelin, nesnelerinin ve diğer dış kaynakların bilgilerini kullanmaktadır. Görsel programlama ile veriler, yeni tasarım çözümleri üretmek ve bir yapının formu ve performansı ile ilgili seçilmiş tasarım kararlarını analiz etmek, değerlendirmek ve optimize etmek için kullanılabilir.

Görsel programlama ile mevcut YBM yazılımları karşılaştırıldığında, örtülü tasarım hedefleri, bir algoritma tarafından temsil edilen tasarım hedefinin nesnel işleviyle açık hale getirilmektedir (Kensek ve Noble, 2014). 2D CAD sistemlerinden YBM

sistemlerine geçişte olduğu gibi, görsel programlama ve verilerle çalışmak değişimle mücadele eden iş yönetimine yeni tutumlar, iş akışları ve uzmanlık gerektirmektedir. Mevcut teknoloji araçlarını bir sonraki seviyeye taşımak için mevcut araçları öğrenmek artık yeterli olmamaktadır. Bunun yerine, tasarım ve inşaat uzmanlarının mevcut araçları kendi ihtiyaçlarına göre şekillendirme ya da kendi araçlarını oluşturma konusunda rahat hissetmeleri gerekmektedir. (Deutsch, 2015).

Günümüzde MMİ endüstrisinde en çok kullanılan görsel programlama ortamı, Rhino için kullanılan Grasshopper aracıdır. Rhino'nun güçlü geometrik özellikleri (Düzensiz rasyonel B-spline'lar gibi) ile birlikte Grasshopper'ın benzersiz işlevlere sahip yüzlerce düğümünden oluşan bir kütüphanesi mevcuttur. Bu avantajlarının yanı sıra Rhino ve Grasshopper yazılımlarının sınırları vardır çünkü tam olarak YBM kavramını akıllı nesneler ile destekleyememekte ve doğrudan bir YBM yazılımı ile birlikte çalışmamaktadır. Böylece Autodesk, görsel programlama ile daha akıllı bir özelleştirme yöntemine ihtiyaç olduğunu fark etmiştir. Autodesk'te Ürün Grup Yöneticisi Matt Jezyk, Grasshopper'ın Rhino'yı kullandığı gibi Revit yazılımına da hesaplama tasarım aracı olarak bir araç geliştirmemiz gerekmektedir, demiştir (Kron, 2014).

Görsel programlama yazılımı olan Dynamo, Ian Keough tarafından açık kaynak projesi olarak geliştirilmiştir. Autodesk, Dynamo topluluğunun büyümesine, projelerin özel ve karmaşık ihtiyaçların karşılanması ve geliştirme sürecinin hızlanması için yazılıma katkıda bulunmaya karar vermiştir.

3.5 Bölüm Değerlendirmesi

Bu bölümde ilgili literatür araştırması yapılmış ve elde edilenler bilgiler değerlendirilmiştir. Elde edilen bulgular Yapı Bilgi Modelleme sistemlerinin proje süreci içerisinde iyi bir temel oluşturmasıdır. Bununla birlikte, iş akışlarında bugün çok fazla manuel adım mevcuttur. Tekrarlayan döngüler, projelerin karmaşıklığı nedeniyle zayıf sonuçlara yol açan el ile yürütülmektedir. Bu bağlamda görsel programlama araçları tasarımcılara ve mühendislere algoritmaları ve yinelemeli iş akışlarını kullanma imkânı vermektedir.

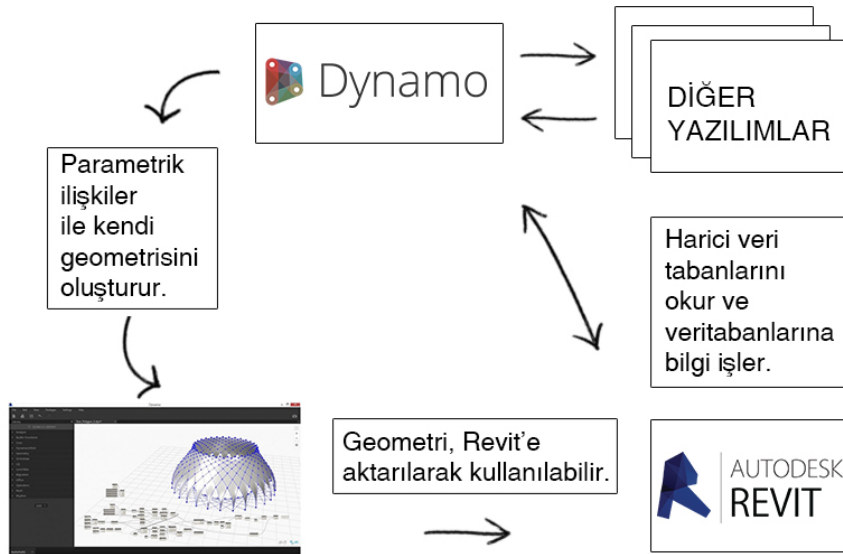
Görsel programlamanın ilkelerini kavrayabilmek için öncelikle algoritmanın tanımı, işlevselliği ve metinsel programlama bu bölümde anlatılmaktadır.

Görsel programlamanın geçmişine bakıldığında, bu araçların MMİ endüstrisinde neden ve nasıl yararlı olabileceği daha açık bir hale getirilmektedir. MMİ endüstrisinde kullanılan görsel programlama dillerinin önemli özellikleri, iş akışları, avantajları ve dezavantajları açıklanmaktadır. Bölüm içerisinde belirtildiği gibi görsel programlamanın avantajları sektör içerisinde yapılması hedeflenen görevler için ağır basmaktadır.

Bölümün sonunda, görsel programlama araçlarının mevcut YBM sistemlerinde yer alan akıllı davranışı genişletmek için neden kullanılabileceği üzerine inceleme yapılmaktadır. Buraya kadar olan kısımda genel olarak görsel programlamanın temelleri üzerinde durulmaktadır. Sonraki bölümde ise görsel programlama diline sahip olan Dynamo aracı daha ayrıntılı bir şekilde ele alınmaktadır.

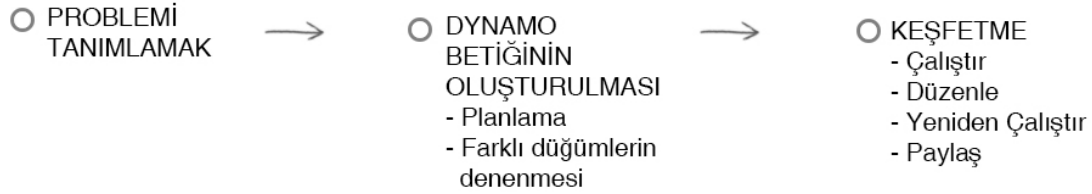
4. DYNAMO İLE GÖRSEL PROGRAMLAMA

Dynamo aracı ücretsiz, kolay ulaşılabilir, görsel programlama dili kullanan açık kaynaklı bir program olup kendinden önce kullanılan görsel programlama aracı olan Grasshopper'den etkilenmiştir (Kensek 2014). Dynamo, yeni fırsatlar oluşturmak için Revit ile işbirliği içerisinde. Dynamo, geometriyi oluşturmak için Revit'teki geometrileri referans alabileceği gibi karmaşık geometrileri noktaya kadar indirgeyerek kendi ara yüzünde de oluşturabilmektedir. Dynamo, DesignScript'e dayalıdır. Bu nedenle harici Dinamik Bağlantı Kitaplıkları (DLL- Dynamic Link Libraries) sınıfları DesignScript sınıfları olarak açığa çıkabilmektedir (Aish ve diğ., 2012). Parametrik bir modelleme motoru olarak, bir seviyede birleştirilebilirlik fonksiyonu ekleyerek Revit'in yeteneklerini genişletmek için tasarlanmıştır (Kensek, 2014). Öncelikle Dynamo iki görevi yerine getirmektedir: kendi geometrisini parametrik ilişkilerle oluşturmakta ve harici veri tabanlarından verileri manipüle edebilmektedir (Şekil 4.1). Ayrıca, Revit API (Application Programming Interface) aracılığıyla Revit veri tabanındaki verilere de müdahale edebilmektedir (Sgambelluri, 2015).



Şekil 4.1 : Dynamo'nun diğer uygulamalarla olan ilişkisi. Kaynak referans alınarak yeniden yorumlanmıştır. (Sgambelluri, 2015)

Sistem içerisinde kullanılan veriler birçok şeye ait olabilir, parametre değerleri, aile geometrisi (family geometry), aile yerleşimi (family placement) gibi. Dynamo ile oluşturulan öğeler Revit'te düzenlenebilir ve parametreler manuel olarak da ayarlanabilir. Oluşturulan nesnelerin bağlantıları korunur ve Dynamo dosyası yeniden çalıştırıldığında öğeler güncellenebilir. Dynamo ile problem çözümünün nasıl olduğuna dair iş akışı Şekil 4.2'de gösterilmiştir.



Şekil 4.2 : Dynamo ile problem çözme iş akışı

Dynamo'nun geliştirici ekibi, kullanıcılar açısından maksimum değer yaratmak için bazı ilkeleri izlemektedirler (Keough, 2015). Bu ilkelerden bazıları aşağıda sıralanmaktadır:

- Kullanıcılar kendi oluşturduğu kod bloklarını kolayca paylaşabilir ve başkaları tarafından oluşturulan düğümleri kullanabilir, genişletebilir veya güncelleyebilir. Böylece Dynamo kullanıcılar tarafından da geliştirilmeye devam etmektedir.
- Her şey bir grafikte gömülmemelidir. Algoritma becerileri ile tasarımcılar ve programcılar tarafından da kullanılabilen ayrı bir platforma da ihtiyaç duyulmaktadır.
- Bir yapıya katkıda bulunan her kullanıcı, Dynamo'nun konseptini ve temel işlevlerini anlayabilmelidir.
- Kullanıcılar grafiklerini kolayca organize edebilmelidir.
- Kullanıcılar tasarımlarıyla gerçek zamanlı olarak analizler yapabilmelidir.

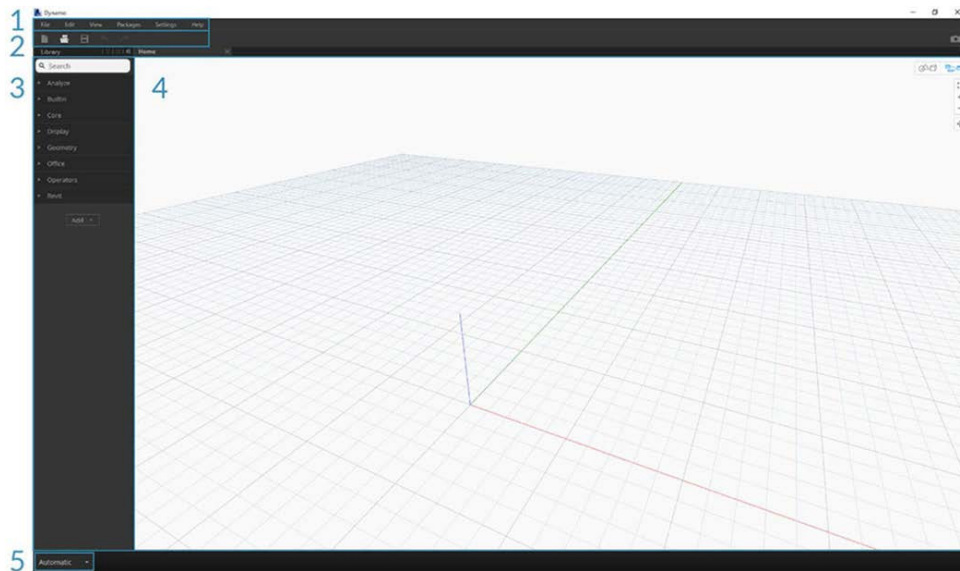
Dynamo için sıralanan prensiplerin çoğu zaten gerçekleştirilmiştir. Bu prensipleri optimize etme süreci devam etmektedir.

Revit yazılımı içerisinde nesnelere ait bazı sınırlamalar içeren formüller oluşturmak her zaman mümkün olmuştur. Görsel programlamanın YBM yazılımına entegre olmasıyla birlikte nesnelerin parametreleri birbiriyle ilişkili olarak ayarlanabilmekte ve karmaşık algoritmalarla genişletilebilmektedir. Dynamo ile birlikte çok fazla zaman

ve emek harcamadan yoğun görevleri otomatikleştirmek mümkündür. Tasarımcıların üretim hızları iş akışı içerisinde Dynamo ile artarken, tasarımcılar günüşiği ve akustik performans kriterleri gibi kriterle bütünleştirilmiş daha fazla tasarım seçeneğini süreç içerisinde keşfetmektedir. Ayrıca Dynamo ile birlikte dijital yapı modellerinde giderek artan veri daha etkin bir şekilde kullanılabilir. Sonuç olarak, Dynamo ile birlikte mevcut YBM proje süreçlerindeki iş akışları optimize edilmektedir.

4.1 Dynamo'ya Genel Bakış

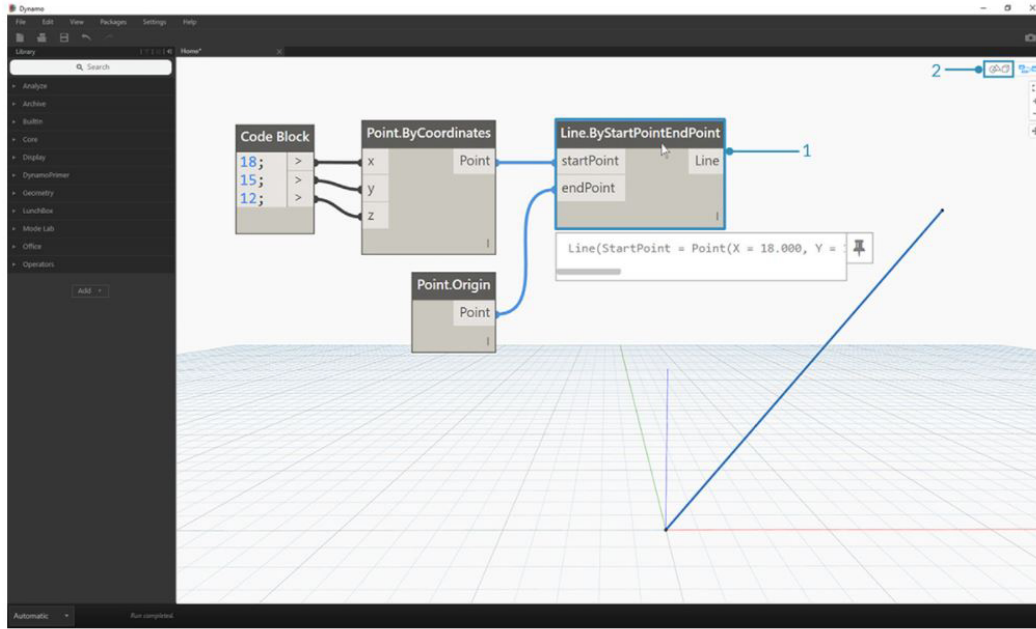
Dynamo ara yüzü kullanıcı için oldukça basit bir şekilde dizayn edilmiştir (Şekil 4.3). Menü çubuğu üst kısımda bulunurken sol tarafta arama çubuğu ve düğüm kütüphanesi bulunmaktadır. Ana pencereye çalışma alanı veya tuval adı verilmektedir. Algoritma oluşturmak için kullanıcının ihtiyacı olanlar elemanlar, düğümler ve düğümleri birbirine bağlayan kablolardır. Alt kısımda ise yer alan çalıştırma düğmesi ile de hazırlanan algoritmanın yürütülmesi sağlanmaktadır. Çalıştırma manuel veya otomatik olarak yapılabilir. Uygulamanın yönü diğer görsel programlama yazılımlarında olduğu gibi soldan sağa doğrudur. Aktif olmayan düğümler açık gri, aktif düğümler gri olarak gösterilirken hata olan düğümler ise sarı renge boyanmaktadır. Bu tarz görselleştirmeler kullanıcıya bir hata durumunda hatanın kaynağının nereden geldiğini kolayca gösterebilmesi açısından önemlidir. Dynamo'da hazırlanan algoritmanın başarılı bir şekilde çalıştırılmasından sonra arka planda oluşan geometriye dair bir ön izleme görüntülenmektedir.



1.Menü 2.Araç Çubuğu 3.Kütüphane 4.Çalışma Alanı 5.Çalıştırma Düğmesi

Şekil 4.3 : Dynamo ara yüzü

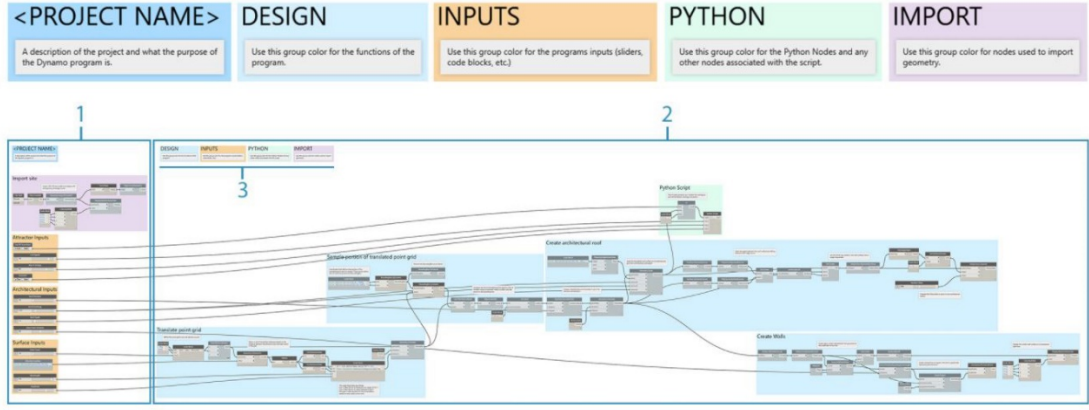
Düğüm, çalışma alanına yerleştirilebilen nesnelerdir (Şekil 4.4). Bunlar, Revit öğelerini, matematik işlevlerini ya da metin tabanlı programlama koduna sahip bir API ile oluşturulabilen ve erişilebilecek herhangi bir şeyi temsil etmektedirler. Düğüm, giriş portları (sol tarafta) ile çıkış portlarına (sağ tarafta) sahiptirler. Giriş ve çıkış noktaları birbirinden farklı veri türlerini temsil etmektedir. Düğüm, ilişkiler kurmak ve bir program akışı oluşturmak için kablolarla birlikte birbirine bağlanmaktadır.



1.Düğüm 2.Geometrinin önizlemesi

Şekil 4.4 : Dynamo’da oluşturulan algoritma ve geometrinin ön izlemesi

Düğüm, mantıksal bir düğüm topluluğuna dâhil ederek gruplandırmak ve grupları ofisin belirlemiş olduğu standartlara göre renklendirmek Dynamo’nun organize edilmesinde oldukça önemlidir. Ofislerde Dynamo kullanımı sırasında farklı düzeylerde görsel programlama becerilerine sahip kullanıcıların, diğer kullanıcılar tarafından hazırlanan Dynamo algoritmasını kullanabilmesi ve algoritmayı anlayarak gerekli değişiklikleri yapabilmesi için oldukça önemlidir. Bu nedenle Dynamo grafikleri ile ilgili belirli standartların geliştirilmesi gereklidir. Dynamo aracı ile oluşturulan algoritmanın karmaşıklığını yönetecek birçok yöntem mevcuttur. Bu yöntemlere örnek olarak düğümün adlandırılması, gruplandırılması, oluşturulan grupların fonksiyonlarına göre farklı renklerle gösterilmesi, grupların hizalandırılması ve gerekli yerlerde düğümlere ve gruplara açıklayıcı notların eklenmesi verilebilir (Şekil 4.5).



Şekil 4.5 : Dynamo grafiğinin düzenlenmesi (Dynamo Primer, t.y.)

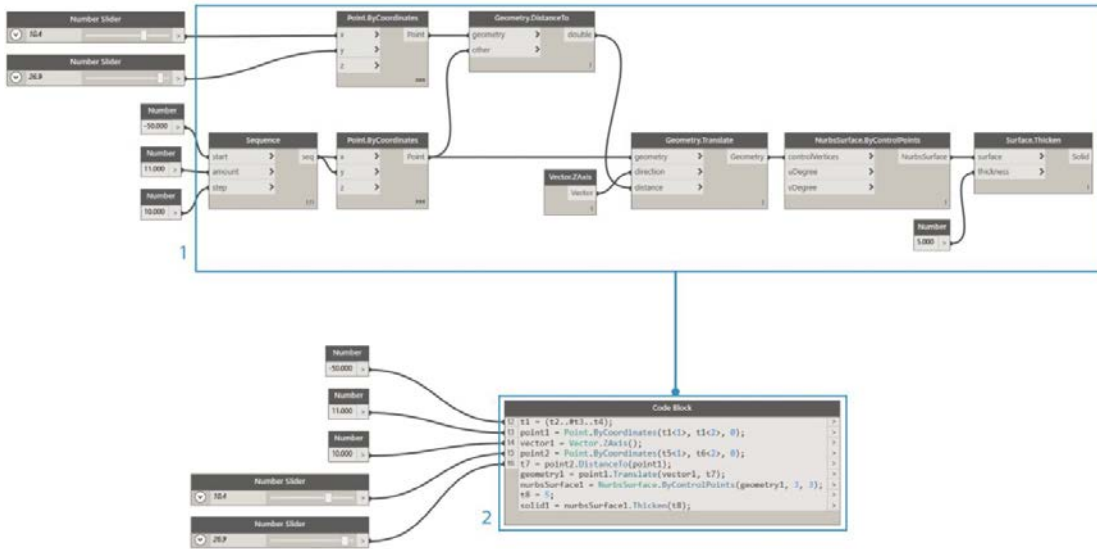
Yukarıdaki şekilde 1 ile numaralandırılmış alanda proje ismi, projeye ait giriş parametreleri ile içeriye aktarılan geometri tanımlanmaktadır. Algoritmanın geri kalan kısmı 2 ile numaralandırılmış alanda gösterilmektedir. 3 ile numaralandırılmış bölümde ise hangi renklerin hangi fonksiyonlar için kullanıldığı gösterilmiştir. Buna göre algoritmaya ait fonksiyonların olduğu düğümler mavi renk ile, proje giriş parametreleri turuncu renk ile, Python programlama dili kullanılarak oluşturulan düğümler yeşil renk ile, Dynamo içerisine alınan geometrik veya non-geometrik veriler mor renk ile ifade edilmektedir. Bu renk stratejileri Dynamo Primer’de bu şekilde belirlenmiştir. Dynamo grafik stratejilerini her ofis kendi ihtiyaçları doğrultusunda düzenleyebilmektedir. Örneğin; White Research Lab firması kendi çalışma standartlarına göre Dynamo grafik stratejilerini hazırlamıştır (Şekil 4.6). Ondejcik’in (2016) açıklamasına göre, bu durum ofiste daha geniş Dynamo kullanımı teşvik etmiş ve tasarım çalışmalarının genel kalitesi ile birlikte ofiste verimlilik artırılmıştır. Şekil 4.5’de yer alan Dynamo Primer’de anlatılan standartlardan farklı olarak Revit ile ilişkilendirilen düğümler ile veri gözetimi ile ilgili düğümler renklendirilmiş ve tamamlanmamış grupların gösterimi için de WIP adı ile ayrıca bir ifade şekli eklenmiştir. Bu stratejiler istenildiği kadar çoğaltılabilmektedir. Sonuç olarak, algoritmanın bütün kullanıcılar tarafından anlaşılabilmesi için proje tanımı, algoritmanın amacı, probleme olan yaklaşımı, oluşturulan düğüm gruplarının tanımı ve fonksiyonlarına göre renklendirilmesi gerekmektedir.

Ayrıca, görsel programlama stratejisi olarak gerekli yerlerde düğümlerin yerine kod blokları (Code Block) kullanılabilmektedir. Bu durum çalışma alanında daha az yer kapladığı için karmaşıklığı azaltmaktadır (Şekil 4.7).



Şekil 4.6 : Dynamo grafiği uygulama planı (Ondejcik, 2016)

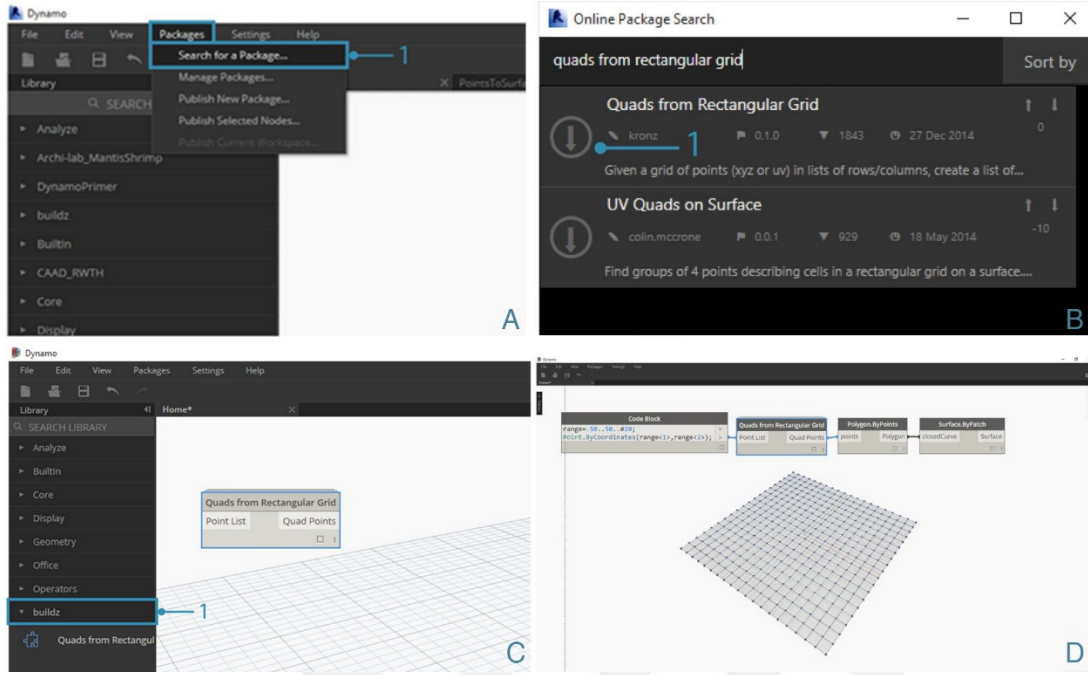
Şekil 4.7’de 1 ile numaralandırılmış alanın içinde yer alan algoritmanın kod bloğu karşılığı 2 ile numaralandırılan alanda gösterilmiştir. Kod bloğu kullanımının grafik görünümü açısından avantajları olduğunu kabul etmek gerekse de farklı düzeylerde olan Dynamo kullanıcılarının hepsine hitap edememektedir. Kod bloğunun bazı kullanıcı grupları tarafından anlaşılması zor olmaktadır.



Şekil 4.7 : Dynamo’da kod bloğunun kullanılması

Düğüm tanımlı bir problem çözmek için bir algoritma oluşturduğunda ve çalışma kontrolleri yapıldıktan sonra düğüm grubu diğer çalışmalarda kullanılmak üzere custom node (özel düğüm) olarak kaydedilebilir. Dynamo’da hazırlanan çoğu algoritma Dynamo kütüphanesinde yer alan mevcut düğümlerle oluşturulabilir. Custom Node yani özel düğüm olarak bilinen düğümler özel problemleri çözmek için geliştirilmiştir. Dynamo’da oluşturulan algoritma içerisinde özel düğümleri (custom node) kullanmak tanımlı problemler için yeniden algoritma oluşturmanın önüne geçtiği için zaman kaybını önlemektedir. Özel Düğümler, paket yöneticisi (Şekil 4.8) içindeki toplulukla paylaşılabilmekte ve yerel olarak saklanabilmektedir. Paket

tanımlanan nokta listesi ile dikdörtgen bir ızgara sistemi oluşturma işlemi için kullanılmıştır (D).



Şekil 4.9 : Dynamo paketlerinin indirilmesi ve kullanılması

Paketler, Dynamo için eklentiler gibidir. Paketler, kullanıcılar tarafından oluşturulmakta ve yerel olarak ofisle veya Dynamo Paket Yöneticisi aracılığıyla bütün Dynamo kullanıcılarıyla paylaşılmaktadır. Paketler, Dynamo algoritmalarında kullanılabilen tek bir düğüm veya düğümler kütüphanesi içerebilir. Paketleri bulmak için en iyi yer, Dynamo içinde paketler menüsünde bulunan Dynamo Paket Yöneticisidir. Kullanılabilir paketlerin açıklamalarına ve eski sürüm erişimlerine göz atmak için paket araması yapılır. Bu menüden kullanılmak istenilen paketler indirilebilir. Bir paketin diğer paketlere bağımlılığı varsa, aynı anda bağımlı paketler de yüklenmektedir. Paket dosyaları “dynamopackages.com” internet sitesi üzerinden elle de indirilebilmektedir, ancak paket yöneticisi kullanıldığında, Dynamo tüm dosya yollarını ve içerik yüklemelerini dikkate almaktadır. En önemlisi de bazı paketler kolayca erişilebilen örnek dosyalar içermektedir. Bu durum kullanıcılara paketteki düğümleri anlama ve kullanma konusunda oldukça yardımcı olmaktadır. Dynamo paketlerinin kullanılması ve geliştirilmelerinin bir diğer nedeni de paketlerin genellikle çok özel sorunları çözmeleridir. Çizelge 4.1’de en çok indirilen Dynamo paketlerinden bazıları yer almaktadır.

Çizelge 4.1: Dynamo Paketlerinden Bazıları

Paket İsmi	Yazarı	Paket Tanımlaması
archi-lab.net	Konrad Sobon	Revit etkileşimi için geliştirilmiş pratik düğümler kümesinden oluşmaktadır.
Lunchbox	Nate Miller	Tekrar kullanılabilir geometri ve veri yönetimi ile ilgili düğümler topluluğudur.
Clockwork	Andreas Dieckmann	Liste yönetimi, matematik ve geometriye dayalı işlemler ile ilgili çeşitli düğümler içermektedir.
Rhythm	John Pierson	Revit verilerinin daha akıllıca yönetilebilmesi için geliştirilmiştir.
SpringNodes	Dimitar Venkov	YBM odaklı iş akışlarını hızlandırabilecek düğümlere sahiptir.
SteamNodes	Julien Benoit	Dynamo iş akışı içerisinde kullanılabilecek çeşitli düğümler kümesine sahiptir.
Rhynamo	Nate Miller	Rhino dosyalarını okumak ve yazmak için kullanılan Dynamo düğüm kütüphanesidir.
Bumblebee	Konrad Sobon	Dynamo'nun Excel dosyalarını okuma ve yazma becerisini geliştiren bir pakettir.
Optimo	Mohammad Rahmani Asl ve Dr. Wei Yan	Optimo, Dynamo kullanıcılarının evrimsel algoritmalar kullanarak sorunları tek ve çoklu hedeflerle optimize etmelerini sağlayan çok amaçlı bir optimizasyon aracıdır.
Mandrill	Konrad Sobon	Dynamo için geliştirilen veri görselleştirme paketidir.
Dynaworks	Adam Sheather	Navisworks için geliştirilen pakettir.

4.2 Dynamo ile Yapılan Uygulamalar

Dynamo'nun esnek yapısından dolayı uygulama alanı oldukça geniştir. Erken tasarım aşamasından proje teslim aşamasına kadar olan süreçte hesaplamalı YBM aracı olan Dynamo, proje süreçlerindeki veri kaybı ve bilgi paylaşımı gibi konularda yaşanan kayıpları en aza indirmektedir. Ayrıca projenin farklı aşamalarında tekrar eden iş yükünü ve bu işler için harcanan zamanı azaltmaktadır.

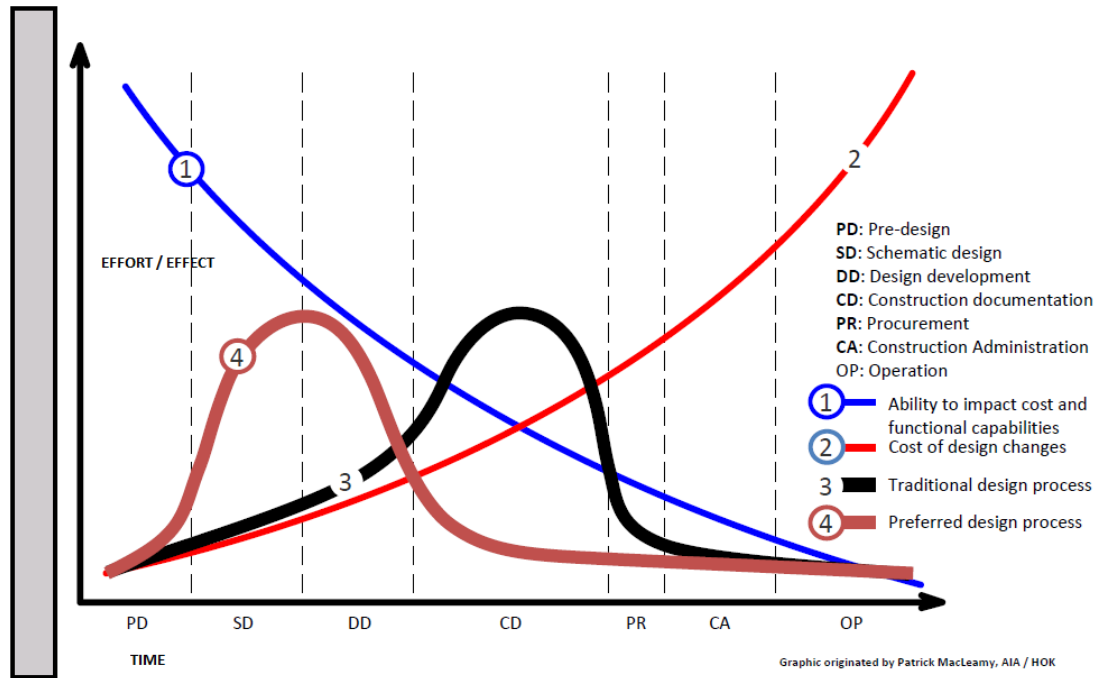
Hesaplamalı tasarım sadece geometrik temsil ile ilgilenmenin yanı sıra nesnelere ve nesneler arasındaki ilişkilere değinmektedir. Nesnelere ve nesne ilişkilerine yeni özellikler eklemek ve bunları manipüle edebilmek de hesaplamalı tasarım ile mümkün olmaktadır. YBM sisteminin geometrik temsilin yanında birçok veri barındırdığı düşünüldüğünde, hesaplamalı tasarımın YBM ile birlikte kullanılmasının verilere müdahale edilebilmesi ile birlikte getirdiği birçok avantaj vardır. Proje ön tasarım aşamasından yapının tesis yönetimi evresinin de dahil olduğu süreçte, özellikle tesis yönetimi için gerekli iç mekân ve çevre kalitesi ile birlikte gerekli bina hizmetleri gibi geometrik olmayan bilgiler giderek daha da önem kazanmaktadır. Verilerin yapı bilgi modelinde proje ve tesis yönetimi için doğru bir şekilde muhafaza edilmesinin önündeki engelleri aşmak için Dynamo gibi hesaplamalı tasarım araçlarının olanaklarından yararlanılmaktadır.

Bu bölümde, projenin erken tasarım aşamasından tesis yönetimine kadar olan süreçte piyasada görülen problemlere Dynamo'nun getirdiği farklı çözüm yaklaşımları değerlendirilmektedir. Dynamo ile birlikte geometrik temsil, mekân planlama, parametrik tasarım, performans ve strüktürel analizler, bina ve model optimizasyonu ve veri paylaşımı gibi alanlarda araştırmalar yapılarak Yapı Bilgi Modelleme ortamına nasıl katkıda bulunduğu incelenmektedir. Dynamo ile yapılan uygulamalar, *analiz ve optimizasyon uygulamaları (bölüm 4.2.1)*, *veri yönetimi ve kalite kontrol uygulamaları (bölüm 4.2.2)* ile *veri alışverişi için harici yazılımlarla oluşturulan entegrasyon (bölüm 4.2.3)* olmak üzere üç ana başlık altında detaylandırılmıştır. Bu başlıklar detaylandırılırken verilen örnekler de Dynamo'dan başka eklentiler de kullanılmıştır. Çünkü Dynamo, paket yöneticisine kullanıcılar tarafından eklenen yeni araçlar ve Autodesk bünyesinde geliştirilen bulut sistemleri gibi yan ortamlarla daha donanımlı hale gelmektedir. Dynamo'nun açık kaynak olması diğer araçlarla birlikte

gelişmesinin önünü açmaktadır. Dynamo'nun gelişmesine katkı sağlayan yan ortamlar ele alınan örneklerle birlikte ilgili bölümlerde anlatılmaktadır.

4.2.1 Analiz ve optimizasyon uygulamaları

Parametrik modellerde ilişkiler, dış ve iç kaynaklı veriler ve kullanıcılar arasında oluşturulabilmektedir. Örneğin, kinetik cephe panelleri güneşe göre tepki gösterebilmektedir. Hesaplamalı tasarım yaklaşımı bu ilişkileri, tanımlanmış kısıtlamalar ile analiz ve optimize edilmiş tasarım çözümleri üretmek için tekrarlayan bir yaklaşımla genişletmektedir. Hesaplamalı tasarımın kavramsal tasarım aşamasında kullanılması ile birlikte erken tasarım evresi dijital yapı modelinin bir parçası haline gelmektedir. Proje sürecinde erken tasarım evresinden itibaren alınması gereken temel kararlar, algoritmalar kullanarak yinelenmeli bir iş akışıyla, oluşturulan kısıtlamalara ve girdilere göre çok erkenden doğru verilere dayalı bir şekilde alınabilmektedir. Tüm kısıtlamalar bir algortmada uygulanabildiğinden, birçok alternatif tasarım seçeneği çok hızlı bir şekilde karşılaştırılabilmektedir. MacLeamy eğrisinde belirtildiği gibi proje için harcanan çabanın daha erken bir tasarım aşamasına doğru kayması proje paydaşlarını zaman ve maliyetten tasarruf ettirmektedir (Şekil 4.10).



Şekil 4.10 : MacLeamy eğrisi ile proje aşamalarında çaba, zaman ve maliyetin karşılaştırması (Url-8)

Dynamo, çeşitli optimizasyon türlerini kullanarak tasarım problemleri ile ilgilenmektedir. Dynamo ile yapılan analiz ve optimizasyon uygulamalarını sıralarsak:

1. Performans tabanlı optimizasyonlar (Günlü optimizasyonları vs.)
2. Akustik optimizasyon
3. Optimo ve NSGA- II optimizasyonları
4. Project Fractal ile yapılan optimizasyonlar
5. Mekân planlama ve optimizasyonu (Space Planning)

Yukarıda sıralanan analiz ve optimizasyon uygulamaları, Dynamo düğümlerini ve yan ortamları tasarım sürecine entegre ederek elde edilmektedir. Yukarıda sıralanan analiz ve optimizasyon türleri yapılan uygulamalarla detaylı bir şekilde ele alınmaktadır.

Dynamo ile performans tabanlı optimizasyon örneği, tez kapsamında araştırmacı tarafından yapılan uygulama ile ele alınmaktadır. Analiz ve optimizasyon başlığı altında incelenen diğer uygulamalar sektördeki Dynamo geliştiricileri tarafından hazırlanmıştır. Araştırmacı, üçüncü kişiler tarafından hazırlanan bu uygulamaların bazılarına ek örnekler geliştirirerek sektördeki kullanım alanını ve sektöre getirdiği olanakları incelemektedir.

4.2.1.1 Dynamo'nun performans odaklı tasarım için kullanılması

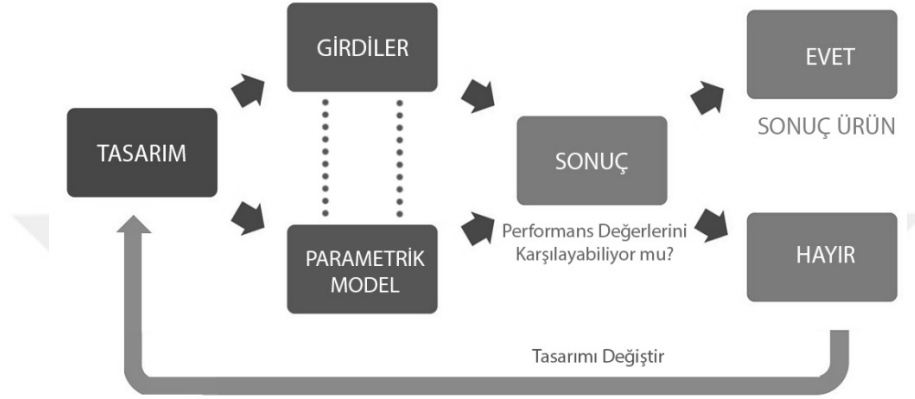
YBM tabanlı yüksek performanslı yapı tasarımı, mimari tasarım disiplinlerinde giderek daha fazla kullanılmaktadır ve kullanıcıların yapı bilgi modellerini verimli bir şekilde üretmelerine ve değiştirmelerine izin vermektedir. YBM sisteminin enerji analiz araçlarıyla entegrasyonu, çoğu zaman hantal ve zor olan enerji simülasyon sürecini büyük ölçüde kolaylaştırma potansiyeline sahiptir. Erken tasarım aşamasında yapı performansını simüle etmek için mimarların yapı bilgi modellerinde mevcut olan geometri, malzeme, inşaat ve teknik sistemler gibi yapı bilgilerine erişmesi gerekmektedir. YBM sistemine gömülü olan bu bilgiler geliştirilen görsel programlama araçları ile birlikte enerji analizi ve optimizasyonu sürecine kolaylıkla dâhil edilmektedir.

Mevcut yapı stoğu ve geleneksel yapım alışkanlıkları gözetildiğinde yapı ve yapıları çevrelerin özellikle enerji verimliliği ve sürdürülebilirlik konularını öne alacak biçimde iyileştirilmesi gerekmektedir. Bu bağlamda örnek kapsamında mevcut bir ofis yapısının cephesinin performans odaklı yenilenmesi konusu ele alınmıştır. Geleneksel ve bütünleşik performans tabanlı tasarım yaklaşımları irdelenerek YBM tabanlı bütünleşik bir sistem önerilmiştir. Bu sistem içerisinde hesaplamalı YBM aracı olan Dynamo ve enerji simülasyon aracı olan Insight 360 gibi programlardan yararlanılmıştır. Cephe yenilenmesinde birincil parametre olarak günışığı sisteme dahil edilmiştir. Dış cepheye ek kabuk olarak oluşturulan ikincil cephede yer alan gölge elemanları ile güneş ışınları arasında doğrudan parametrik ilişki oluşturulmuştur. Gün ışığının iç ortama kontrollü bir şekilde alınmasıyla birlikte ofis yapısının iç ortam koşullarının karşılaması gereken aydınlatma şiddeti ve kamaşma düzeyinin yüksek olması ve aşırı ısınma gibi problemlerin önüne geçilebilmektedir.

Odağında enerji verimliliği olan sürdürülebilir bir bina tasarımında bütünsel tasarım yaklaşımı ile birçok alternatif senaryo test edilerek optimum çözüme ulaşılmaya çalışılmaktadır. Enerji performansı optimizasyonu süreci kompleks bir süreç olup, içerisinde yapı ve çevre ile ilgili birçok parametre ve bu parametrelerin birbiriyle olan ilişkilerini barındırmaktadır. Bu noktada tasarım girdilerini, girdiler arasındaki ilişkileri tanımlayabilmeyi ve elde edilen farklı tasarım alternatiflerinin arasından enerji performansı açısından optimum olan çözümün seçilebilmesine olanak tanıyan birçok tasarım aracı (EnergyPlus, Ecotect, Equest vs.) mevcuttur. Ancak, bu enerji optimizasyon araçları bina ve bina alt elemanları ile enerji verileri arasında parametrik ve eş zamanlı değişime olanak tanıyan bir ilişki ortaya koyamamaktadır. Bu bağlamda enerji optimizasyonu için bütünleşik tasarım süreci ve yenilikçi tasarım metodolojileri gerekmektedir. YBM ortamı bize enerji optimizasyonu için gerekli olan verileri sağlamaktadır. Hesaplamalı YBM aracı olan Dynamo ile birlikte tasarım parametreleri ve parametrelerin birbiriyle kurduğu ilişkiler değiştirildiğinde yeni tasarım alternatifleri ve bu alternatiflerin enerji kullanım değerleri oluşturulabilmektedir. Bu veriler doğru bir şekilde yönetilir ve enerji optimizasyonu sürecinde kullanılırsa enerji değerlerinin ölçüleceği modele veri oluşturmak için harcanan zamandan ve emekten tasarruf etmek mümkündür. Bu durumda geleneksel enerji optimizasyonu için oluşturulan iş akış şeması (Şekil 4.11), yerini daha bütünleşik bir iş akış şemasına (Şekil 4.12) bırakmaktadır.



Şekil 4.11 : Geleneksel performans odaklı tasarım iş akış şeması



Şekil 4.12 : Bütünleşik performans odaklı tasarım iş akış şeması

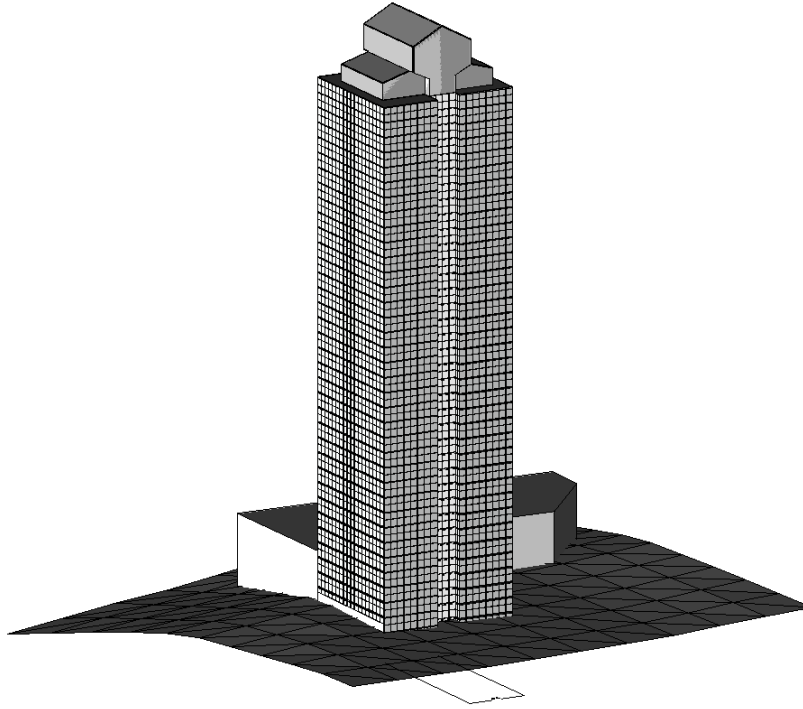
Bu uygulama kapsamında literatür araştırmasından elde edilen bilgiler ile birlikte daha önce yapılmış olan çalışmalar dikkate alınarak hesaplamalı tasarım metodu ile performans tabanlı bir cephe tasarımı gerçekleştirilmesi hedeflenmiştir. Çalışma için mevcut bir ofis yapısı seçilmiştir. Mevcut yapı olarak ofis yapısının seçilmesinin en büyük nedeni, ofis yapılarının diğer yapılara göre en yüksek enerji tüketimi miktarlarına sahip olmasıdır. Ofis yapılarında enerjinin çoğunluğu ısıtma, soğutma, havalandırma ve aydınlatma için harcanmaktadır. Ofis yapılarının karşılaması gereken iç ortam performans kriterlerinin neler olduğu üzerine araştırma yapılmıştır. Bu araştırmaların sonucunda iç ortam performans kriterlerini önemli ölçüde etkileyen faktörün aydınlatma düzeyi olmasından dolayı cephe tasarımında birincil öneme sahip parametre olarak gün ışığı sürece dahil edilmiştir. Günışığının bina içerisine alınması ve iç mekânda oluşturduğu etkiler cephedeki doluluk-boşluk oranına, cephede kullanılan elemanların konumlarına, yüzeydeki cam malzemenin özelliğine ve eğer gölgeleme elemanı varsa bu elemanların malzemesine, konumuna ve yönelimine bağlı olarak değişkenlik gösterebilmektedir. Bu nedenle mevcut yapının aydınlatma analizleri yapılmıştır. Bu analizlerin sonucunda olması gereken iç ortam performans kriterlerine ulaşmak için nasıl bir cephe yenilemesinin yapılması gerektiği üzerine

düşünülmüştür. Gün ışığı ile birlikte kullanıcıların görsel konforlarının sağlanması için kamaşmaya ve aşırı miktarda doğrudan ısı kazanımına karşı çeşitli gölgeleme elemanları incelenmiştir. Sonuç olarak seçilen mevcut yapıya ek olarak gün ışığı performansına dayalı ikincil bir cephe tasarlanması önerilmiştir. Cephe yenilemesi ile hedeflenen amaç binanın özgün tasarımında öngörülemeyen performans kriterlerine ulaşılmasını sağlamaktır. Performans odaklı cephe yenilemesi kapsamında birçok alternatif senaryoların, tasarım ve simülasyon araçları kullanılarak mevcut binaya uygulanması ve bunun sonuçlarının değerlendirilmesi hedeflenmektedir. Binaya sonradan adapte edilebilecek alternatiflerin ısıtma ve soğutma yükleri ve aydınlatma yükünü azaltmadaki etkilerini gözlemlemek hedeflenmiştir. Mevcut yapının dış kabuğuna sonradan uygulanabilecek alternatifler olarak, farklı güneş kontrol elemanlarının kullanılması seçilmiştir. Uygulama için sırasıyla gerçekleştirilen adımlar:

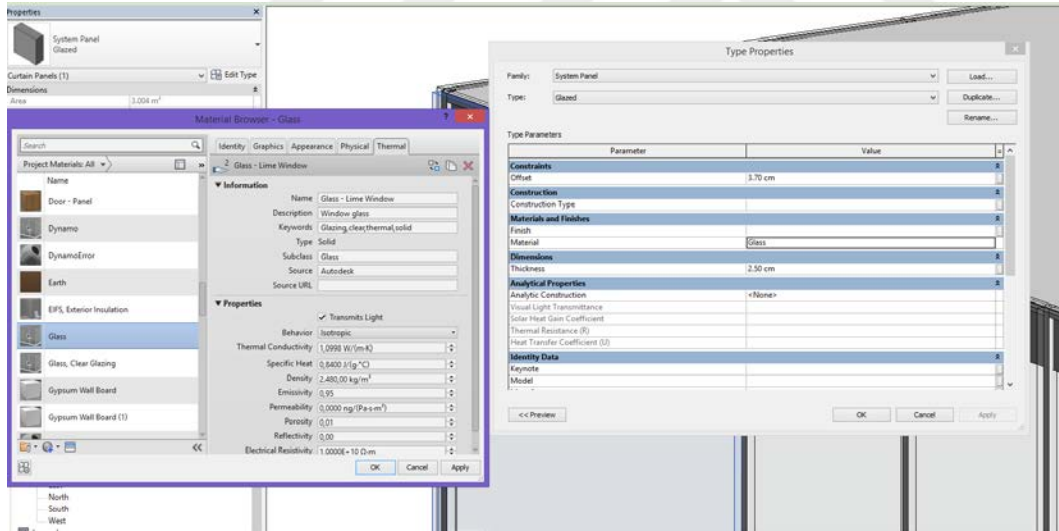
1. Mevcut binanın seçilmesi
2. Mevcut binanın modellenmesi- (Revit Architecture)
3. Mevcut binanın enerji performans değerlerinin hesaplanması- (Insight 360)
4. Mevcut binanın cephe yenilenmesi ile ilgili hedeflerin ve parametrelerin belirlenmesi
5. İkincil cephenin oluşturulması- (Dynamo)

Binanın Seçilmesi: Performans tabanlı tasarım yaklaşımını örneklemek için İstanbul Maslak'ta bulunan 38 katlı ofis yapısı olan Beybi Giz Plaza seçilmiştir.

Mevcut Binanın Modellenmesi: Beybi Giz Plaza, Revit Architecture programında modellenmiştir (Şekil 4.13). Dış cephe, renkli alüminyum giydirme cephe ve kompoze levha kaplamadır. Isıtma ve soğutma sistemi fan-coiller ile sağlanır. Kazanlar tarafından ısıtılmış ya da chiller grubu tarafından soğutulmuş su fan-coillerden geçerken üfleme yapılır. Mevcut yapıya ait giydirme cephenin malzeme özellikleri ile birlikte kullanılan havalandırma sistemi Revit'in veri tabanına işlenmiştir (Şekil 4.14). Daha sonra bu veriler enerji simülasyonu sırasında kullanılmıştır.

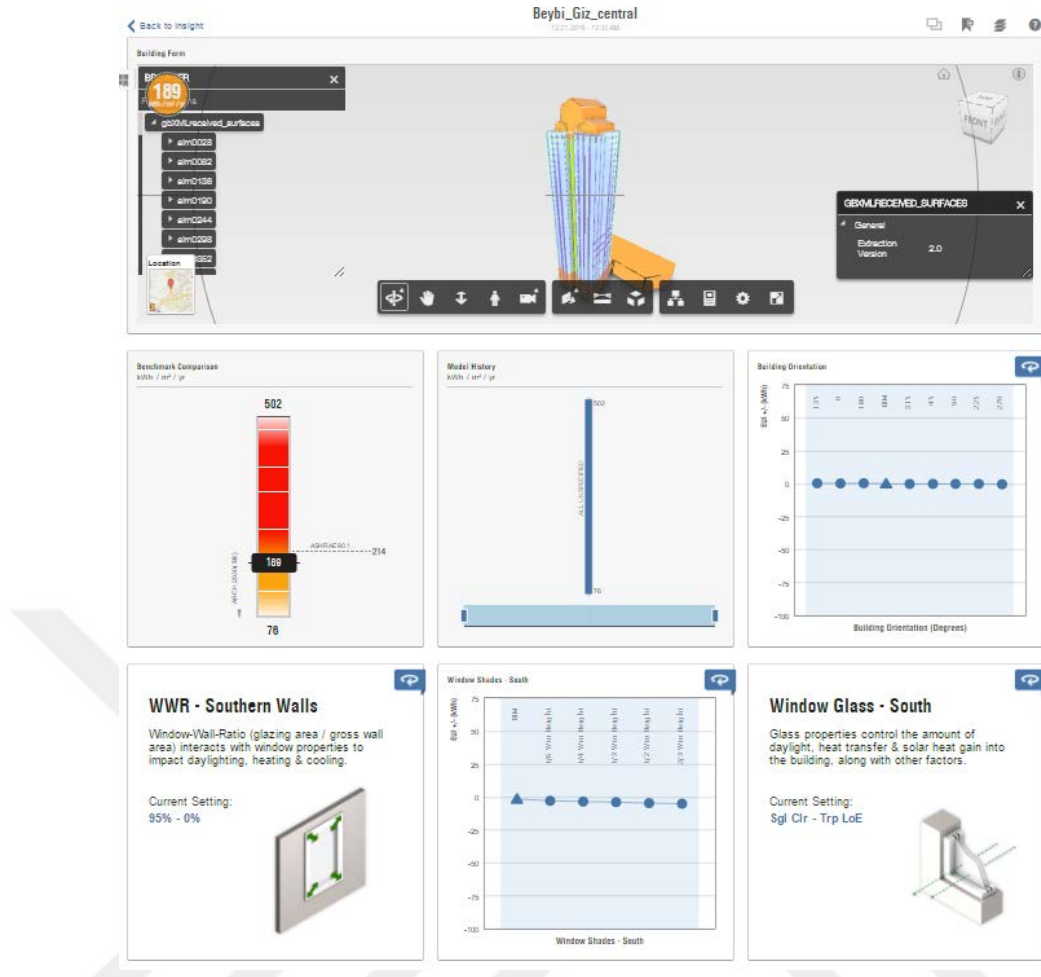


Şekil 4.13 : Beybi Giz Plaza Revit modeli



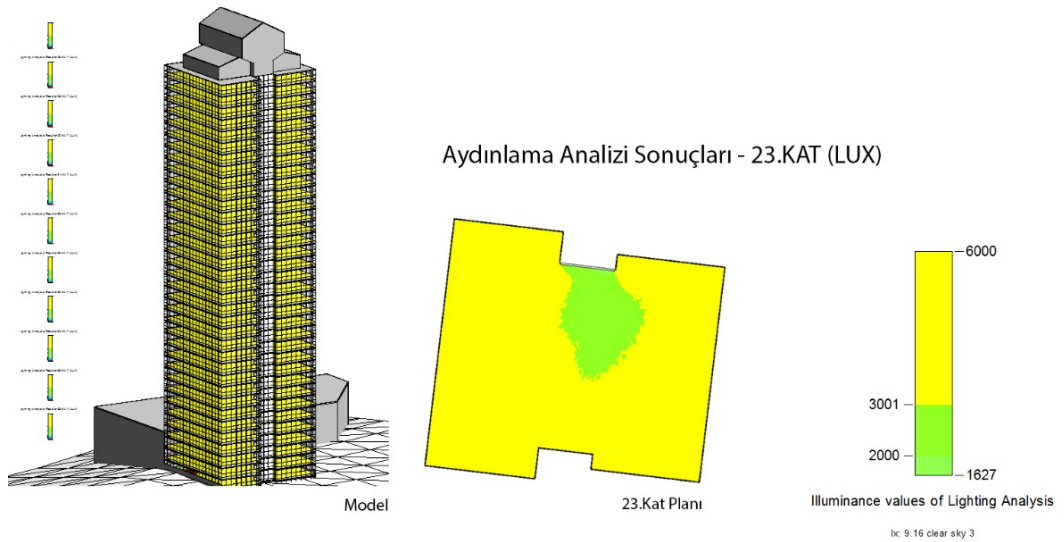
Şekil 4.14 : Mevcut giydirmce cephe özelliklerinin Revit'e işlenmesi

Mevcut Binanın Enerji Performans Değerlerinin Hesaplanması: Bina enerji simülasyonu Insight 360 programı ile yapılmıştır (Şekil 4.15). Insight 360, Revit modelleri için tam dinamik termal simülasyon gerçekleştirdiğinden dolayı enerji simülasyon programı olarak seçilmiştir. Analiz çıktıları olarak elde edilen enerji maliyet aralığı sonuçları kullanıcılara rehberlik ederek bilinçli tasarım kararları vermeye yardımcı olmaktadır.



Şekil 4.15 : Insight 360 programında yapılan enerji analizi çıktıları

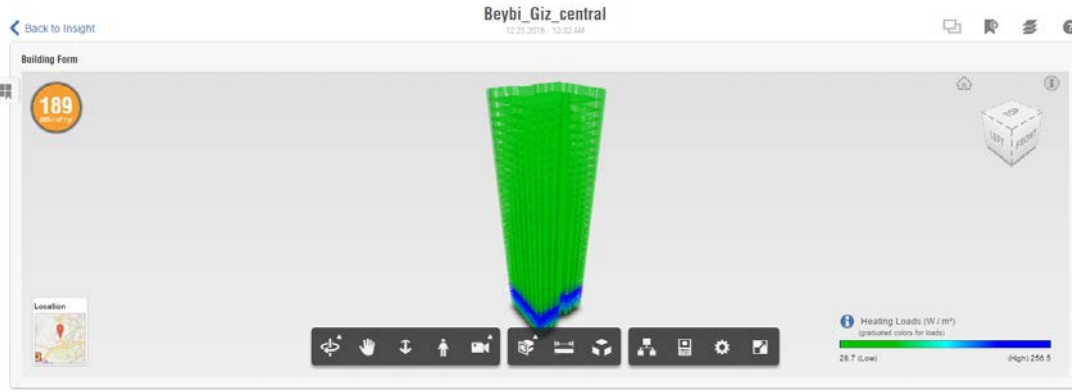
Aydınlatma Analizi: Yapılan aydınlatma analizine göre ofis katlarındaki aydınlık düzey sarı bölgelerde 3001-6000 lux değerlerindiyken yeşil bölgelerde ise 1627-3001 lux değerleri arasındadır (Şekil 4.16).



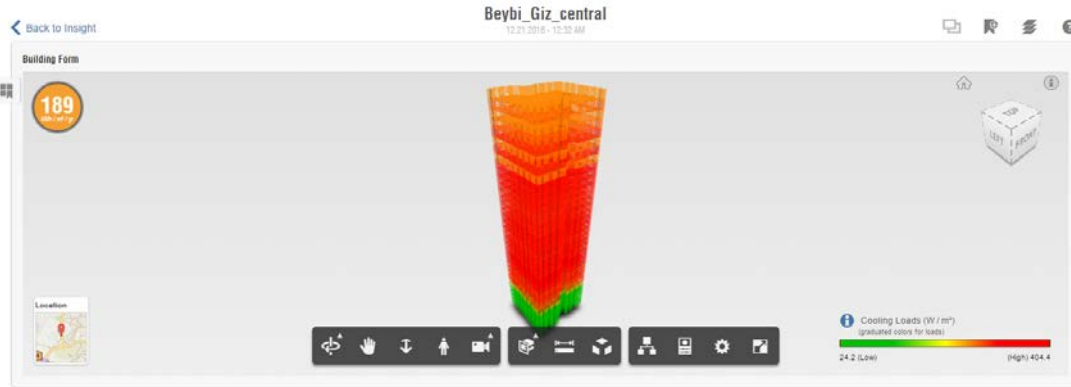
Şekil 4.16 : Aydınlatma analizi çıktıları

TS EN 12464 nolu “Işık ve Işıklandırma- İş Mahallerinin Aydınlatılması- Bölüm 1: Kapalı Alandaki İş Mahalleri” standardında belirtilen işyerlerindeki bazı alanlarda ve işlerde gerekli aydınlatma şiddeti değerlerine göre ofis yapılarında istenen aydınlık düzeyi 500 lux olarak belirtilmiştir. Mevcut yapının aydınlatma analiz değeri bu değerin oldukça üstündedir. Tasarlanacak olan ikincil cephe ile iç mekandaki aydınlık düzeyin istenilen seviyelere düşürülmesi amaçlanmıştır.

Isıtma ve Soğutma Yükleri Analizi: Insight 360’da yapılan analiz sonuçları Şekil 4.17 ve Şekil 4.18’de gösterilmiştir.



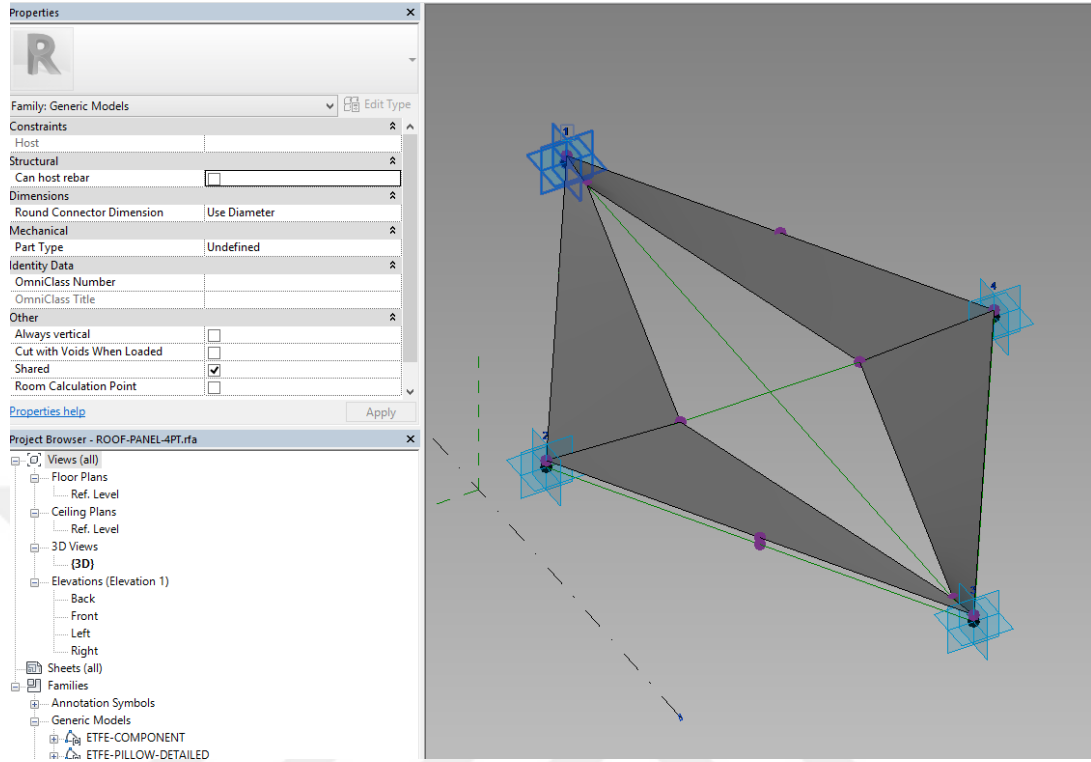
Şekil 4.17 : Insight Beybi Giz Plaza ısıtma yükleri analizi



Şekil 4.18 : Beybi Giz Plaza soğutma yükleri analizi

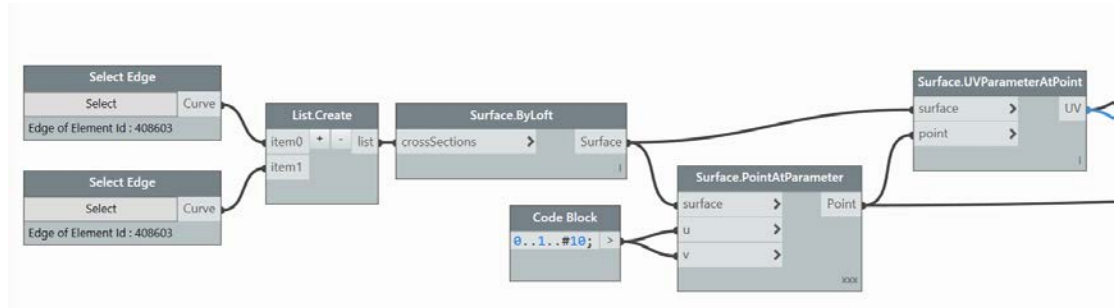
Mevcut Binanın Cephe Yenilenmesi İle İlgili Hedeflerin ve Parametrelerin Belirlenmesi: Mevcut binanın enerji analiz sonuçlarına göre binaya gün ışığının daha kontrollü bir şekilde alınması için ikincil bir cephe tasarlanmasına karar verilmiştir. Bu cephede yer alması düşünülen gölge elemanlarının boyutlarının güneş ile doğrudan ilişkilendirmesi düşünülmüştür. Revit ortamında uyarlanabilir cephe elemanı (adaptive component) olarak tasarlanan elemanın iç yüzeyinde 4 tane adapte edilebilen

nokta oluşturulmuştur (Şekil 4.19). Bu noktaların cephe yüzeyindeki yeri güneşe göre konumlandırılacaktır.



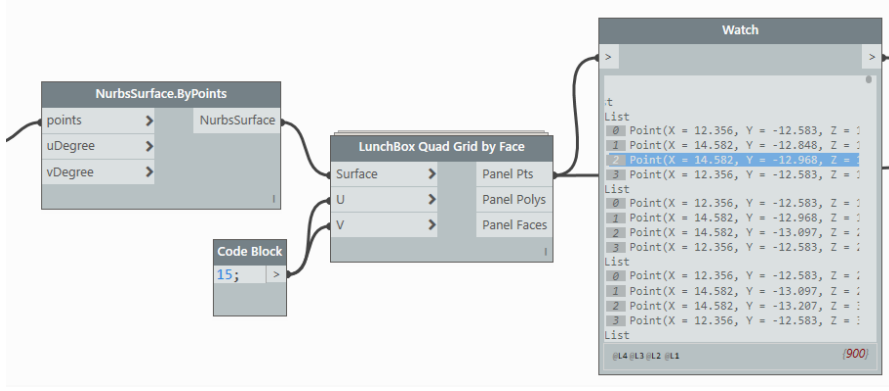
Şekil 4.19 : Güneşe bağlı olarak boyutları değişebilen gölge elemanı

İkincil Cephenin Oluşturulması: İkincil cephe için öncelikle Revit programında oluşturulan kütle(mass) üzerinden iki kenar seçilerek yüzey oluşturulmuştur. Yüzey üzerinde belli aralıklarla noktalar tanımlanmıştır. Yüzey üzerinde oluşturulan bu noktalar parametrik olarak tanımlanan yüzey için kontrol noktaları olarak kullanılmıştır (Şekil 4.20).



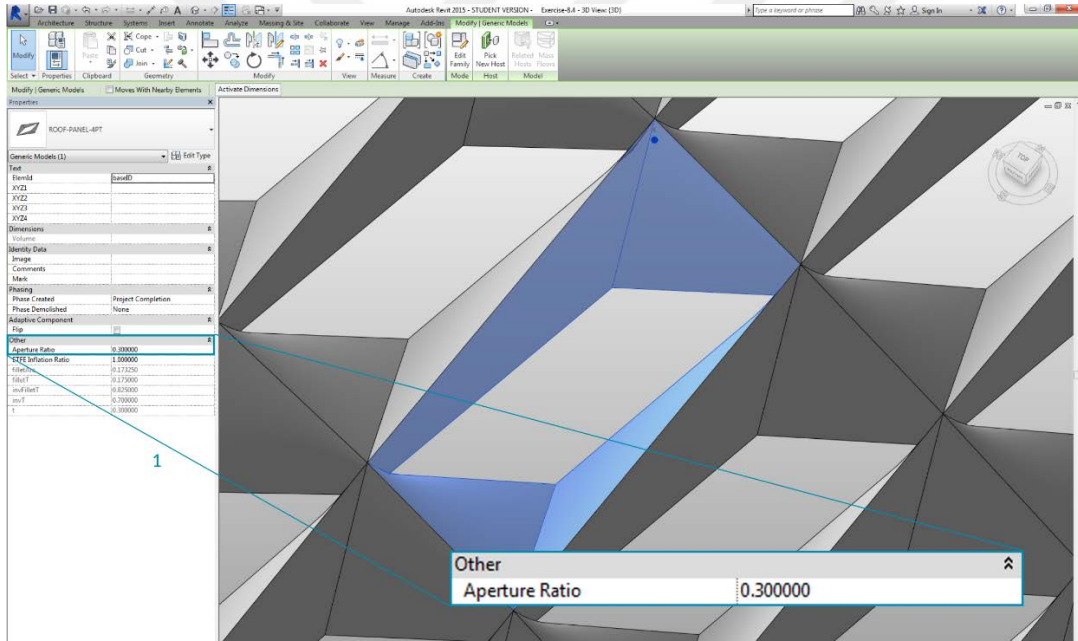
Şekil 4.20 : Yüzeyin oluşturulması ve noktaların tanımlanması

Yüzey oluşturulduktan sonra yüzey üzerinde dört noktalı uyarlanabilir bileşenleri (cephe panelleri) sıralamak için yüzey panelizasyonu gerçekleştirilmiş ve bunun için Dynamo'nun içerisine indirilen LunchBox paketi kullanılmıştır. (Şekil 4.21)

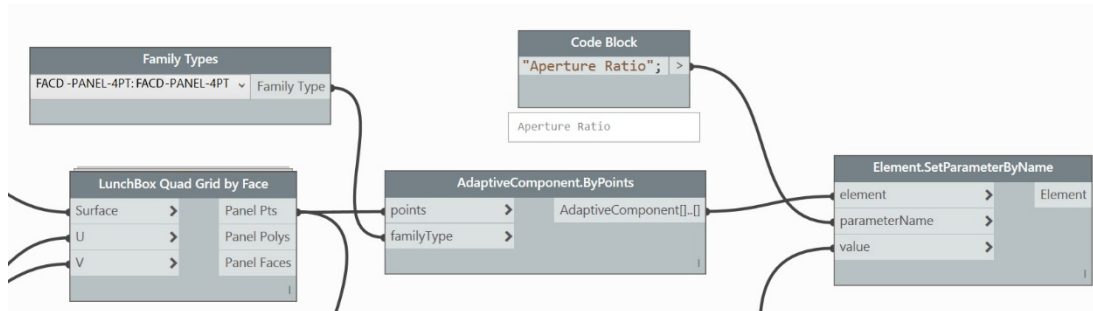


Şekil 4.21 : Yüzeyin panelizasyonu

Yüzeyde panelizasyon işlemi gerçekleştirildikten sonra bu panellere yerleştirilecek olan gölge elemanları Dynamo ortamına alınarak, bu gölge elemanlarının merkezlerindeki boşluk, açıklık oranı (Aperture Ratio) parametresine (Şekil 4.22) bağlanmıştır (Şekil 4.23).

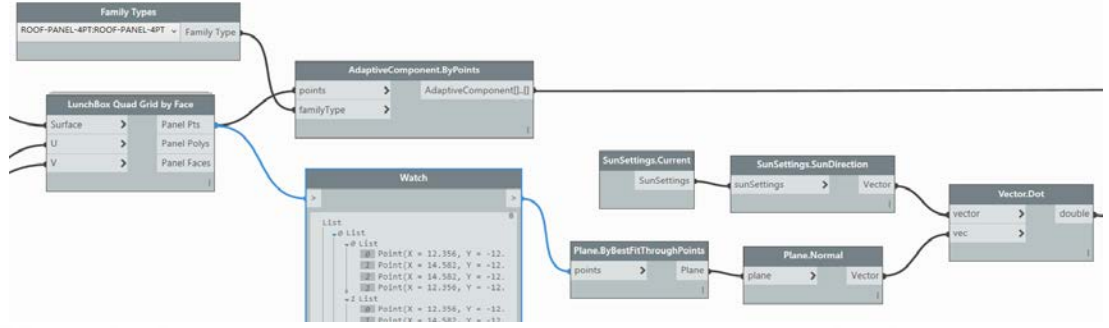


Şekil 4.22 : Cephe elemanında tanımlanan açıklık oranı (aperture ratio) parametresi



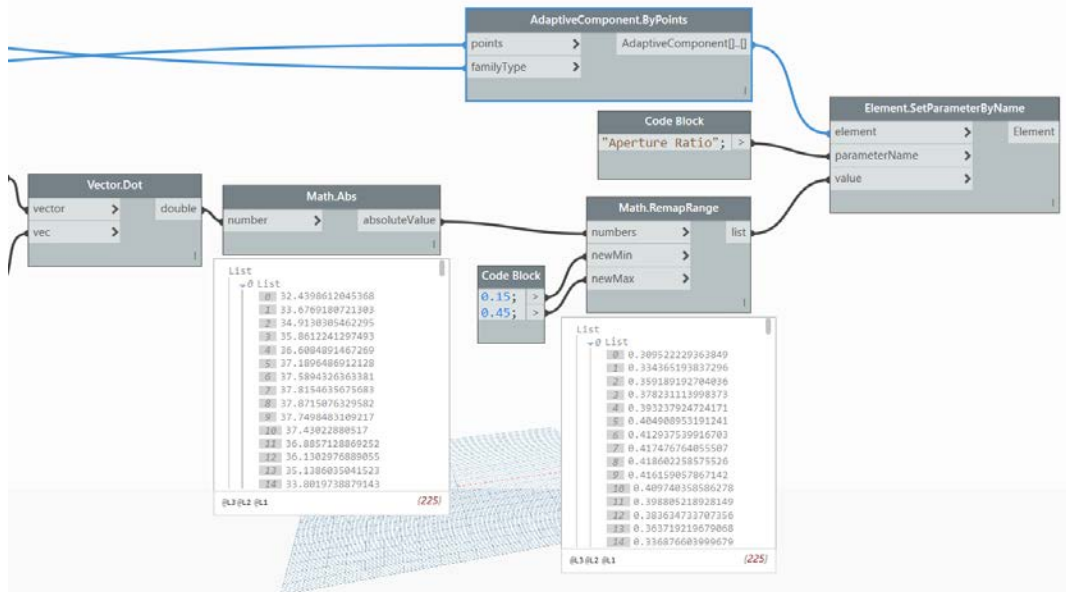
Şekil 4.23 : Gölge elemanlarının yüzeye yerleştirilmesi ve açıklık oranı parametresinin tanımlanması

21 Haziran saat 13.00 zaman diliminde güneşten çıkan ışınlar Dynamo ortamına “SunSettings.SunDirection” düğümüyle alınarak bu ışınlar birer vektör olarak tanımlanmıştır. LunchBox paketi ile oluşturulan panellerin normalleri ile ışın vektörlerinin arasındaki mutlak açı değeri elde edilmiştir. (Şekil 4.24).



Şekil 4.24 : Panel normallerinin ve güneş ışınlarının (vektörlerinin) arasındaki mutlak farkın bulunması

Yapılan işlemten sonra elde edilen mutlak değerler geniş bir aralıkta yer almaktadır. Cephede yer alacak olan gölgeleme elemanlarının açıklık oranını belli bir değerde tutabilmek için yeniden eşleme (remap) işlemi yapılmıştır. Yeniden eşleme işlemi aldığı girdi listesini belirli sınırlar arasında yeniden tanımlayarak gölge elemanlarına güneşe olan konumuna göre 0.15 ve 0.45 açıklık oranı arasında değerler atanmıştır (Şekil 4.25).

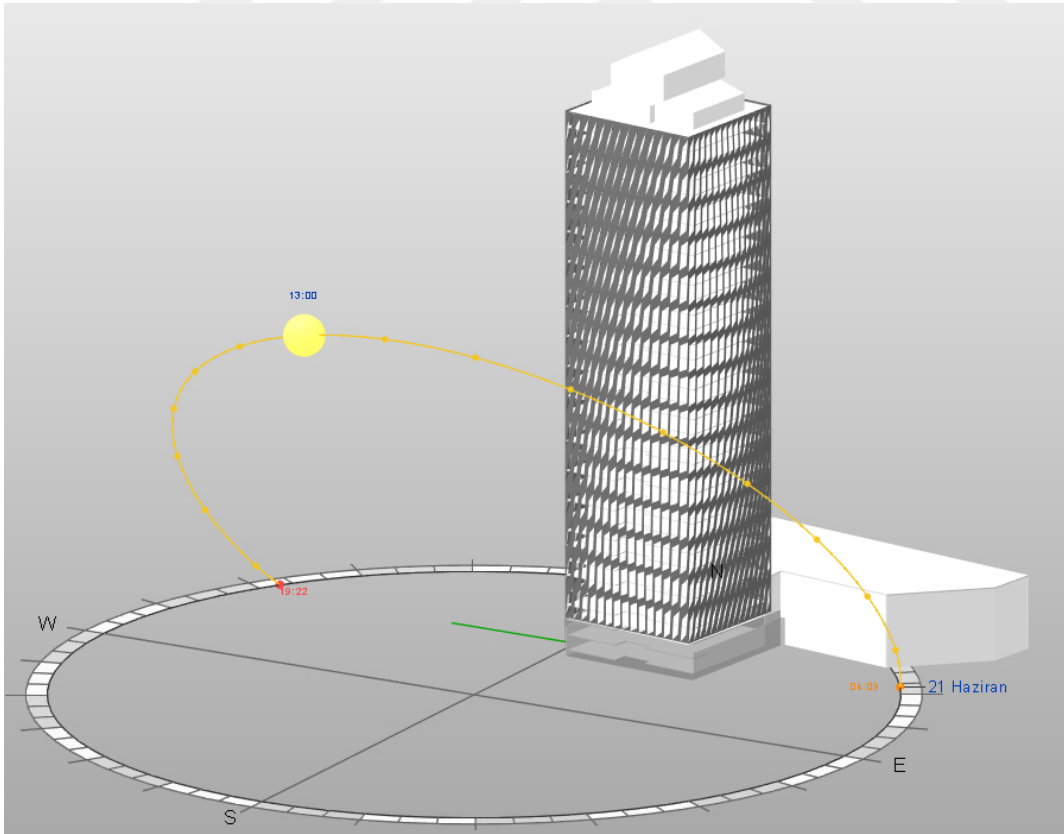


Şekil 4.25 : Gölge elemanın açıklık oranını belli değerler arasında tutmak için yapılan yeniden eşleme işlemi

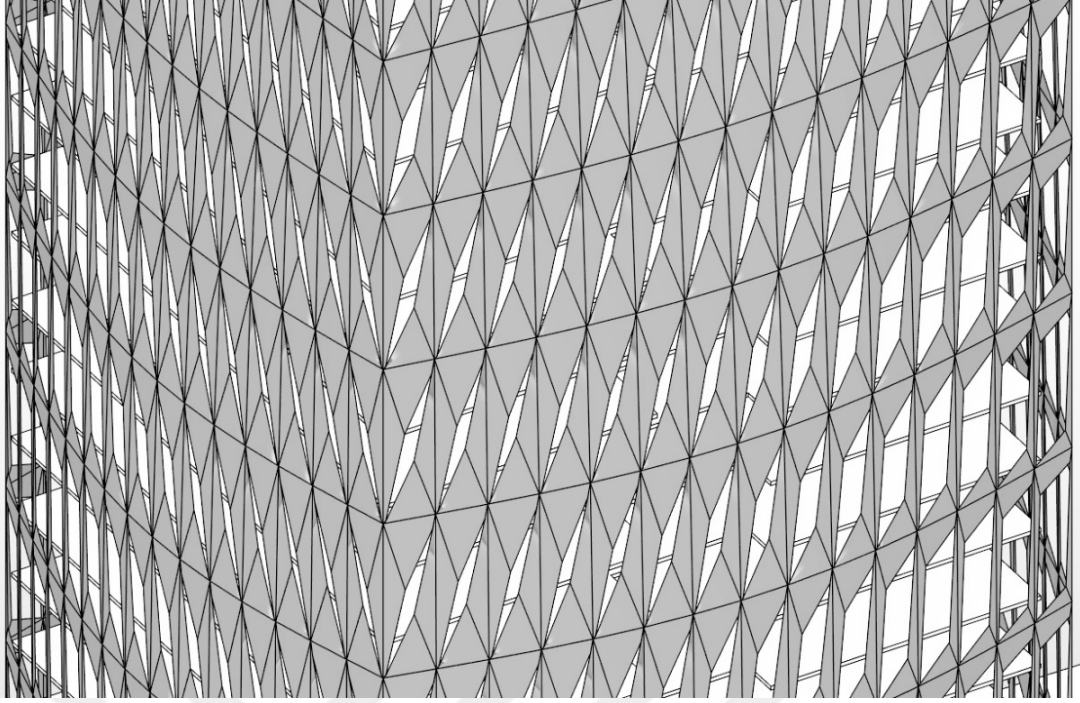
Son olarak, “Element.SetParameterByName” düğümü (Şekil 4.25) ile gölgeleme elemanlarının akıllı nesnesinin ne olacağı, açıklık oranının hangi değerler arasında olacağı ve bu oranın Revit ortamında nasıl isimlendirileceği tanımlanmıştır.



Şekil 4.26 : Dynamo’da oluşturulan algoritmanın tamamı



Şekil 4.27 : Dynamo ile oluşturulan ikincil cephenin Revit ortamında görüntülenmesi



Şekil 4.28 : İkincil cephedeki gölgeleme elemanlarının açıklık oranlarının, güneşe göre farklılaşması

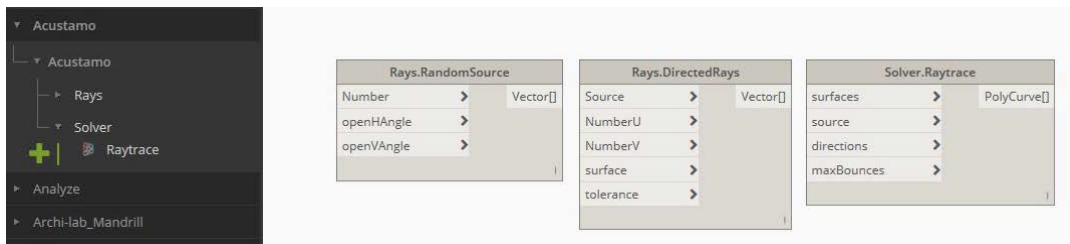
Binaların enerji tüketimi ile çevrelerine olan etkilerinden dolayı tasarımcılar, özellikle tasarımın erken aşamalarında, enerji performansını optimize etmenin yollarını aramaktadırlar. Enerji optimizasyon çalışmalarının erken tasarım aşamasında yapılması, tasarımcıyı projenin ilerleyen süreçlerinde yönetemeyeceği karar değişikliklerinden kurtarmaktadır. Bu durumda projenin ilerleyen süreçlerinde karar değişikliği yapmanın zorlaşması ve yapılan her değişikliğin proje maliyetinin artmasına sebep olduğu unutulmamalıdır. Tasarımcı proje sürecinin başında doğru kararlar verdiğinde bu kararların üzerine doğru bir şekilde ilerleyebilmektedir.

Binaların enerji performansında yer alan birçok parametre nedeniyle, enerji performans optimizasyonu süreci kompleks bir süreç olmaktadır. Optimizasyon sürecinde yer alan parametreler malzeme özellikleri, geometri, hava durumu verileri, kullanıcı davranışları ve benzerleri olabilmektedir. Bu parametreleri değiştirerek oluşturulan farklı tasarım alternatifleri test edilerek sonuç ürün optimize edilmeye çalışılmaktadır. Optimize edilmeye çalışılan konulara örnek olarak, gün ışığından elde edilen ısı akışının enerji kazanımları ile evin konumlandırılmasını yönlendirmek veya yenilenebilir enerji tarafından toplanan ısı kazanımlarını depolamak olabilir. Enerji analiz ve optimizasyon sürecinin karmaşıklığı ve performans araçlarının eksikliğinden dolayı tasarımcılar, tasarımdaki enerji performansını proje tasarım sürecine dahil

etmeyebilir ya da enerji tasarım uzmanlarından yardım almayı seçebilirler. Enerji uzmanları simülasyonlar yaparken mimari model, enerji modeline dönüştürülmek yerine, enerji modeline aktarılmaktadır. Bu durum farklı programların birbiriyle çalışmasının sonucu olarak karşımıza çıkabilen veri ve zaman kaybına sebep olurken hata eğilimli de olabilmektedir. Oysa, YBM ve YBM tabanlı hesaplamalı tasarım aracı olan Dynamo'nun süreç içerisinde birlikte kullanılması tek bir model üzerinde analiz ve optimizasyonların yapılmasına olanak tanımaktadır. Bir YBM modeli, enerji analizi için gereken verilerin çoğunu içermekte ve verilerin Dynamo'ya aktarımı için herhangi bir hazırlık gerekli olmamaktadır. Böylece enerji analizleri sırasında meydana gelen veri ve zaman kaybı azaltılmaktadır. Ayrıca, enerji analiz ve optimizasyonları için “Energy Analysis for Dynamo”, “Ladybug”, “Honeybee” gibi Dynamo paketleri geliştirilmiştir. Dynamo için geliştirilen bu paketlerle birlikte parametrik enerji modeli oluşturularak enerji analizi için gereken iş akışları oluşturulabilmektedir.

4.2.1.2 Dynamo'nun akustik optimizasyon için kullanılması

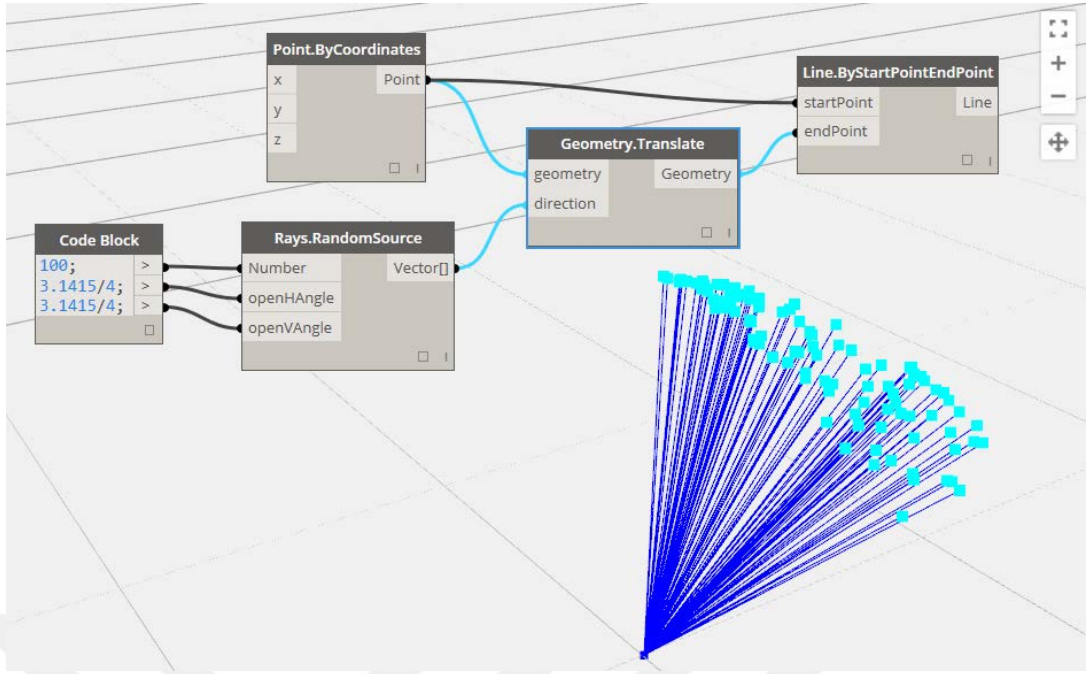
Dynamo ortamında akustik optimizasyon için kullanılmak üzere Andrea Vannini tarafından Acustamo adlı bir paket geliştirilmiştir. Bu paket de üç tane düğüm bulunmaktadır (Şekil 4.29).



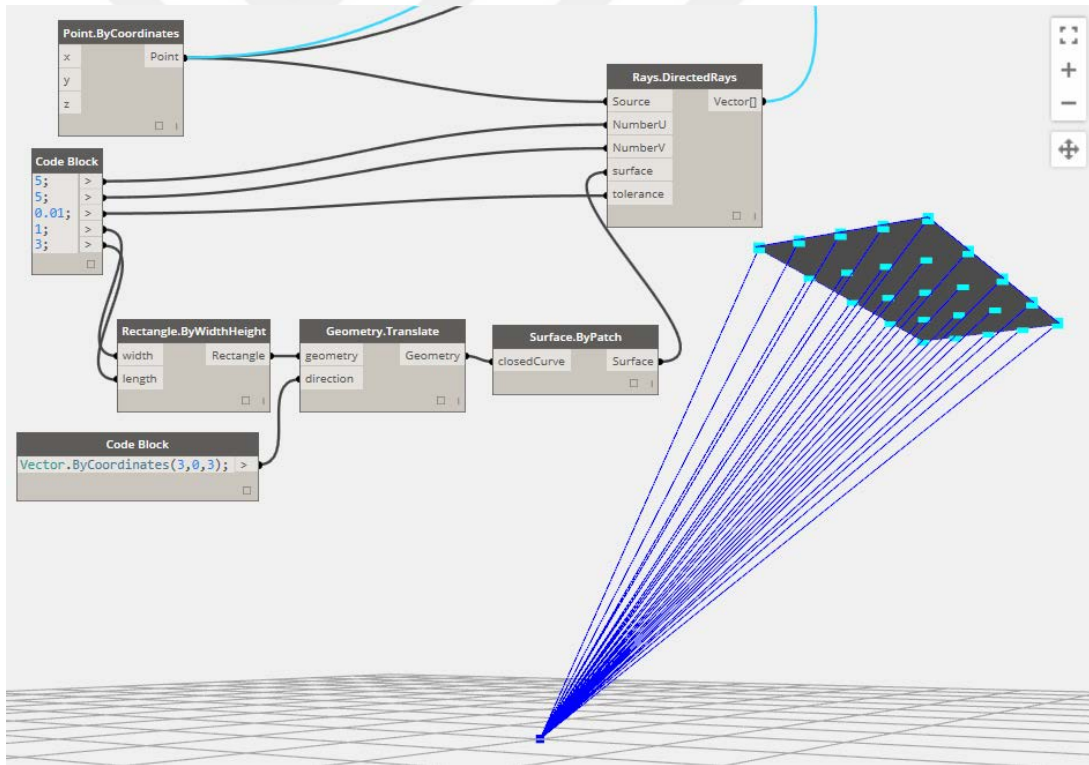
Şekil 4.29 : Acustamo paketine ait düğümler

Acustamo paketi ile akustik analiz yapılırken ses kaynağını nokta olarak sisteme tanıtmak gerekmektedir. Ses kaynağından çıkan ışınlar vektör olarak temsil edilmektedir. “Rays.RandomSource” düğümünün kullanıldığı algorithmada bu ses ışınlarının nasıl temsil edildiği görülebilmektedir (Şekil 4.30).

Ses ışınları oluşturulduktan sonra sesin çarptığı yüzeyde nasıl bir etkisi olduğunun analiz edilebilmesi için Acustamo paketine ait “Rays.DirectedRays” düğümü kullanılmaktadır. Akustik panelleri oluşturmak için bu düğümü kullanmak elverişlidir. Şekil 4.31’de yer alan örnekte ses ışınlarının bir yüzeye nasıl çarptığı görülmektedir.

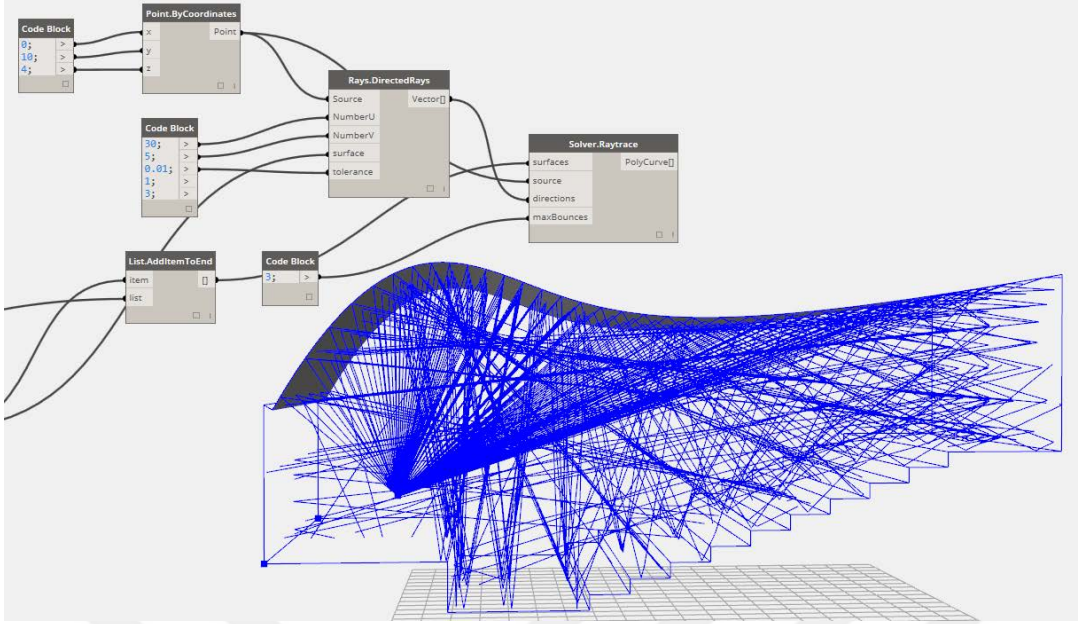


Şekil 4.30 : Ses kaynağının temsil edilmesi (Vannini, 2015)



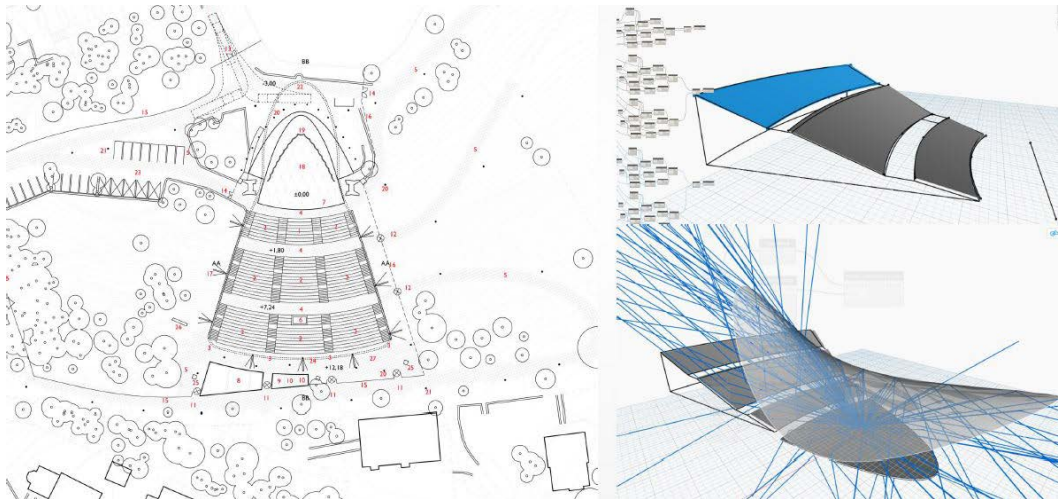
Şekil 4.31 : Ses ışınlarının yüzeyde temsil edilmesi (Vannini, 2015)

Acustamo paketi içerisinde yer alan bir diğer düğüm olan “Solver.Raytrace” düğümü ile ses ışınlarının çarptığı yüzeylerden elde edilen yansımalar Dynamo ortamında görselleştirilerek hesaplanmaktadır (Şekil 4.32).

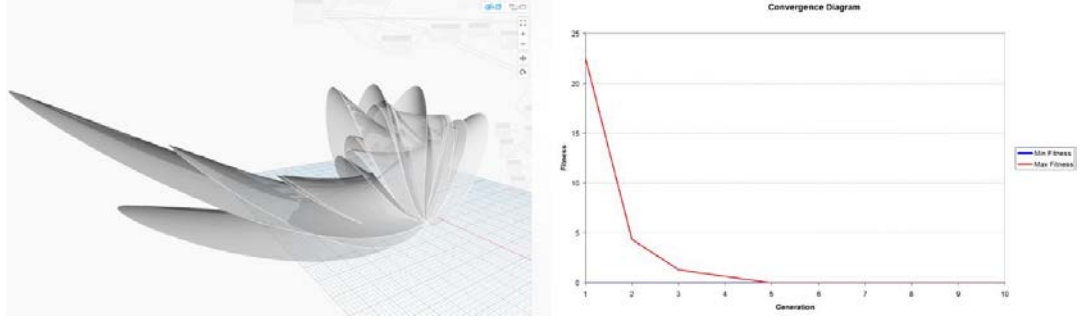


Şekil 4.32 : Ses ışınlarının yüzeyde temsil edilmesi (Vannini, 2015)

Projenin erken tasarım aşamasında, tasarımcı, tanımlanmış bir akustik performans değerini karşılamak için tasarladığı yapının formunun nasıl olması gerektiğine karar verebilmek için birtakım analiz ve optimizasyonlara ihtiyaç duymaktadır. Hudson ve Strathaus 2015 yılında Autodesk University’de yapmış oldukları konuşmada akustik optimizasyonun kullanılmasına iyi bir örnek vermişlerdir. Polonya’da tasarlanan performans alanının formunun oluşturulması için Dynamo’da yer alan Acustamo ve Optimo paketlerinden (bölüm 4.2.1.3) yararlanılmıştır. Acustamo paketi ile ses ışınları ve sesin çarptığı yüzeyler Dynamo ortamında oluşturulmuştur (Şekil 4.33). Optimo paketi ile de tasarım alternatifleri için optimizasyon süreci gerçekleştirilmiştir (Şekil 4.34). Süreç sonucunda elde edilen tasarım şekil 4.35’de yer almaktadır.



Şekil 4.33 : Dynamo Acustamo paketi ile ses ışınlarının ve yüzeylerinin oluşturulması (Hudson ve Strathaus, 2015)



Şekil 4.34 : Dynamo Optimo paketi ile tasarım alternatiflerinin denenmesi (Hudson ve Strathaus, 2015)



Şekil 4.35 : Sonuç tasarım (Hudson ve Strathaus, 2015)

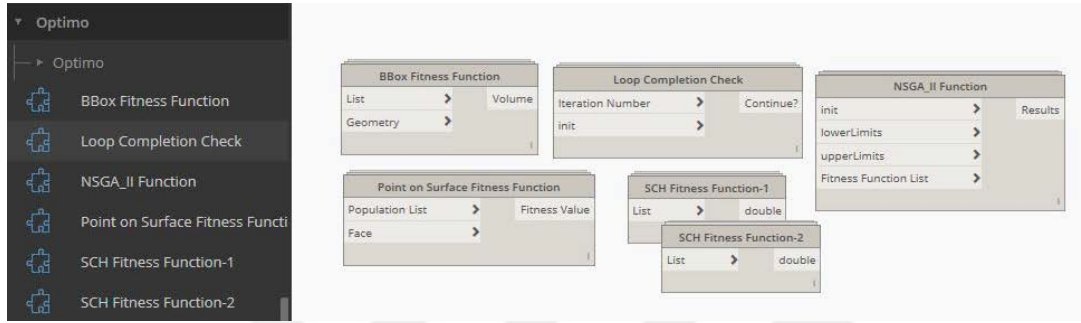
Dynamo Acustamo paketi piyasada yer alan akustik analizler ile ilgili problemlere çözüm getirmektedir. Fakat şu anda ki Acustamo paketindeki yapılan hesaplamalarda malzeme bilgisi sürece dahil edilmemiştir. Acustamo paketinin bu yönünün geliştirilmesi ilerisi için önemli olmaktadır.

4.2.1.3 Optimo ve NSGA-II optimizasyonları

YBM, mimari, yapısal, enerji, akustik, aydınlatma vb. gibi çoklu performans kriterlerinin analizi için kullanılabilecek modeli entegre bir veri tabanı olarak temsil etmektedir. Tasarımcının yüksek performanslı bir bina tasarımına doğru ilerlemesine yardımcı olmak için çok kriterli optimizasyon ile entegre edilecek uygun bir tasarım aracıdır. Çok kriterli optimizasyon süreci, YBM'nin parametrik özelliklerinden ve yapı bilgi modelinde bulunan verilerden yararlanmaktadır. Bununla birlikte, YBM tabanlı bina tasarımı optimizasyonu ile ilgili araştırmalar oldukça sınırlı kalmıştır (Asl ve diğ., 2015a).

Gözlemlenen boşluğa cevap olarak, Çok Amaçlı Optimizasyon (Multi Objective Optimization-MOO) için YBM tabanlı bir görsel programlama paketi olan Optimo

geliştirilmiştir. Böylece tasarım sürecinde bina performansı ile ilgili çok disiplinli optimizasyon yapılmasına olanak tanınmıştır. Optimo, YBM tabanlı optimizasyon için Autodesk Revit ile parametrik olarak etkileşime geçmek için geliştirilmiş, açık kaynaklı, Dynamo içine kurulabilen çok amaçlı bir optimizasyon paketidir (Şekil 4.36). Optimo, Mohamad Rahmani Asl ve Dr. Wei Yan tarafından, Texas A&M Üniversitesi BIM-SIM laboratuvarı bünyesinde Java'da Metaheuristik Algoritmaların C # uygulamasını sağlama amacı ile jmetal.NET açık kaynak koduna dayanılarak geliştirilmiştir.



Şekil 4.36 : Dynamo Optimo paketi düğümleri

Optimo, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) temel alınarak çalışmaktadır. NSGA-II, çok amaçlı hesaplmalı bir genetik algoritma türüdür. İki veya daha fazla kriterin optimizasyonunun amaçlandığı problemlerin çözümü için kullanılmaktadır. Çoklu kriterli olan problemlerin optimizasyonları için genel olarak iki farklı yaklaşım söz konusudur:

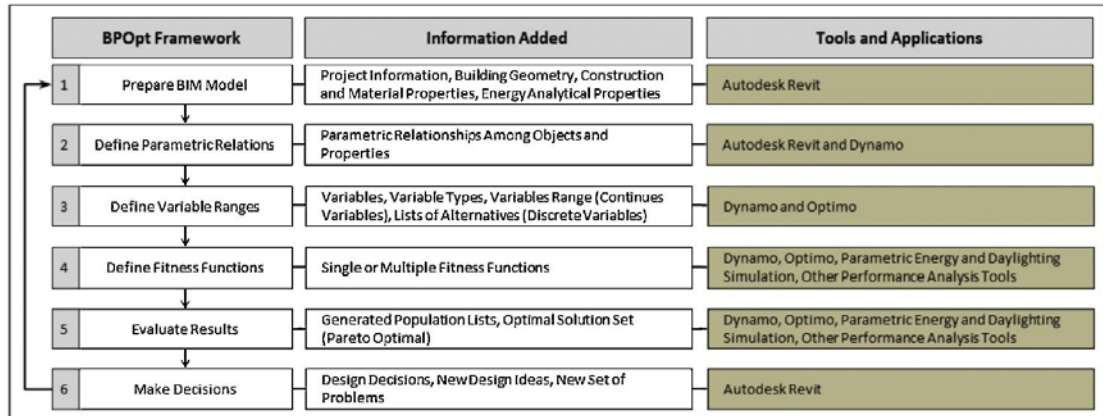
1. Basit Kümeleme (Simple Aggregation)
2. Optimo Pareto (Pareto Optimal)

Basit kümelemede genel bir birleştirici formül, alt bireysel formülleri kümelemek için kullanılır. Bu formül ağırlık çarpanları belirleyerek kurulabilir. Her zaman mümkün olmamakla birlikte, birleşik fonksiyon belirlenirken, alt fonksiyonların birbirleri ile ilişkisi ve ağırlıkları da belirlenmelidir (Asl ve diğ., 2015a).

Optimum Pareto küme yaklaşımında ise birden fazla hedef için optimum sonuçlar kümesi bulunur. Aynı derecede en uygun çözümleri bularak, en azından bir başka bireysel hedefin daha da kötüye gitmesine neden olmaksızın, tek bir bireysel hedefi iyileştirmeyi mümkün kılarak karar vermeyi desteklemektedir (Asl ve diğ.,2015a).

Asl ve diğerlerine göre (2015b), bina performansını optimize etmek ve tasarım kararları verilirken geri bildirim almak için YBM tabanlı performans optimizasyon uygulama süreci (Şekil 4.37) altı ana adımda açıklanabilmektedir:

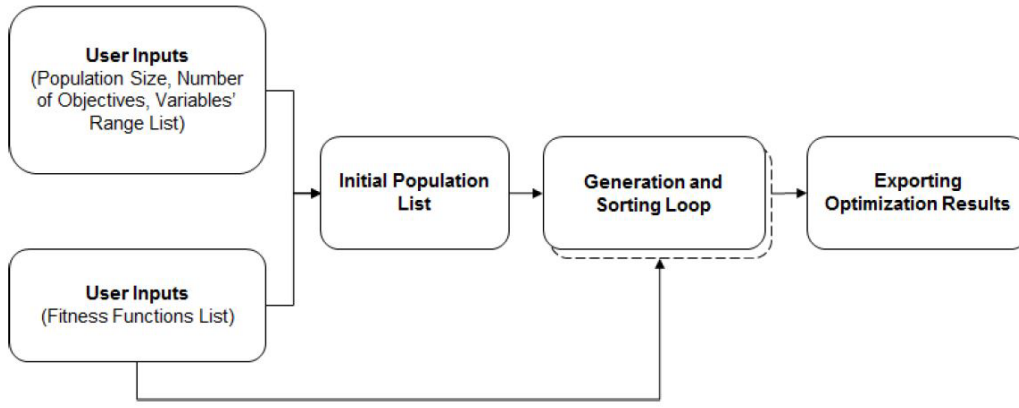
1. Yapı bilgi modelinin, performans analizi için yapı nesnelere ait gerekli analitik özellikleri uygulanarak hazırlanması.
2. Yapı nesneleri arasında parametrik ilişkilerin tanımlanması.
3. Optimizasyon süreci için karar değişkenlerinin ve alanlarının tanımlanması.
4. Optimizasyon işlemi için fitness fonksiyonlarının tanımlanması.
5. Değişkenler ve fitness fonksiyonlarını optimizasyon paketine besleyerek optimizasyonun yapılması.
6. Karar verilmesi – (sadece tasarımcılar tarafından ele alınabilecek ana tasarım görevinin yerine getirilmesi)



Şekil 4.37 : YBM tabanlı performans optimizasyonu uygulama süreci (Asl ve diğ., 2015b)

YBM tabanlı performans optimizasyonu süreci, tasarımcıların karar vermelerini ve modelin performansı üzerindeki etkilerini test etmelerini sağlayan yinelemeli bir döngü olarak tasarlanmıştır. Sürecin son adımında, tasarımcı sonuçları değerlendirir ve ilerlerken iki yolu kullanabilir:

1. Süreç içerisinde sağlanan en uygun çözüm setinden bir tasarım alternatifi seçilir ve tasarım ilerler.
2. Sonuçlara göre tasarımcı optimizasyon ayarlarında ve parametrik ilişkilerde değişiklik yapar ve istenen tasarım elde edilene kadar aynı işlemi tekrarlar.



Şekil 4.38 : Optimo çalışma yapısı (Asl ve diğ., 2015a)

Şekil 4.38’de gösterildiği ve aşağıda tarif edildiği gibi, Optimo beş ana bölüme ayrılmaktadır (Asl ve diğ., 2015a):

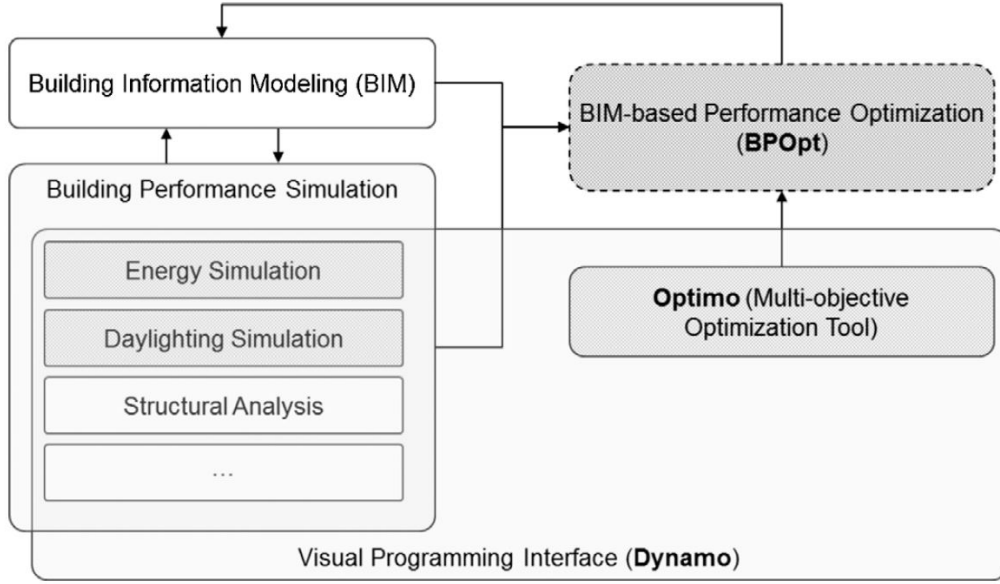
- **Kullanıcı Girdileri (Nüfus Boyutu, Hedef Sayısı ve Karar Değişkenleri):** Bu kısım, kullanıcı girdisini optimizasyon algoritmasının ve karar değişkenlerinin aralıklarının özelliklerine göre toplar. Hedeflerin sayısı, optimizasyon sürecine dahil edilen fitness fonksiyonlarının sayısını tanımlar. Tasarım parametreleri sürekli değişkenler (alt ve üst sınırlarla tanımlanmış), ayrık değişkenler (değişkenlerin listesi olarak tanımlanır) veya her ikisi olabilir.
- **Kullanıcı Girişleri (Fitness Fonksiyonları Listesi):** Optimo, performans simülasyonu fonksiyonlarını fitness fonksiyon argümanları olarak optimize etmek için tasarım sürecinde kullanılabilir. Bu nedenle, kullanıcı, optimizasyon kaynak kodunda herhangi bir değişiklik yapmak zorunda kalmadan fitness fonksiyonlarını harici fonksiyonlar (Dynamoda özel düğümler kullanarak) olarak ekleyebilir. Aksi takdirde çok fazla deneyim ve programlama uzmanlığı gerektirebilir.
- **İlk Nüfus Listesi:** İlk rastgele nüfus listeleri ve bunların uygunluk değerleri Optimo yapısının bu bölümünde üretilir. Optimo, tanımlanan etki alanında rastgele karar değişkenleri oluşturmak için değişken aralıkları kullanır, bu değişkenleri kullanarak tasarım seçeneklerine yönelik fitness işlevlerini hesaplar ve uygunluk değerlerini nüfus listesine atar.
- **Üretim ve Sıralama Döngüsü (Generation and Sorting Loop):** Bu, Optimo’da kullanıcı tarafından tanımlanan tamamlama kontrolüne ulaşana kadar

yinelenen ana optimizasyon döngüsüdür. Döngü, kullanıcı girdilerinin yanı sıra ilk çözüm listesini alır ve çapraz popülasyonu oluşturur.

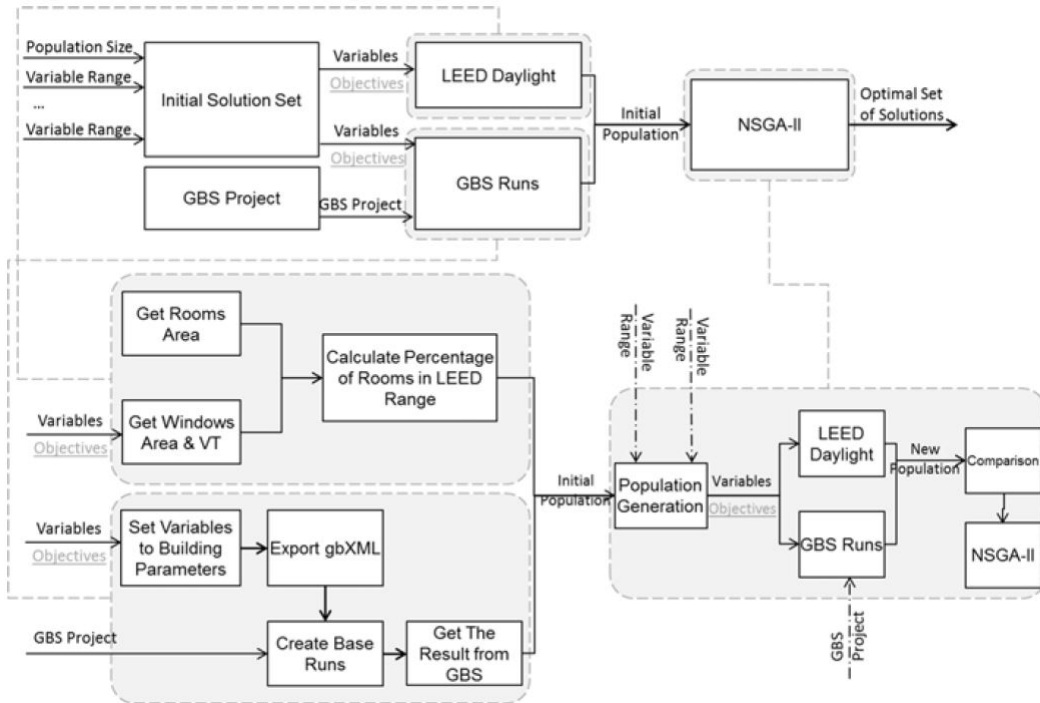
- Optimizasyon Sonuçlarının Dışa Aktarılması: Tüm yinelemelerin ve son Pareto Optimal kümelerinin popülasyonu, daha fazla analiz için CSV (Comma Separated Values) formatında dışa aktarılabilir. Pareto Optimal Set, set içindeki her bir çözüm için, en az bir başka hedefin, değişimden önceki durumundan daha kötüye gitmesine neden olmadan tek bir amacı iyileştirmenin mümkün olmadığı, eşit derecede optimal çözümler içerir.

Görsel programlama ve YBM entegrasyonu, sürdürülebilir bina tasarımına büyük ölçüde fayda sağlayabilecek bir yöntem olarak kullanılmaktadır. Parametrik modelleme, değişkenleri ve onların ilişkilerini değiştirerek bir tasarım alanının yaratıcı keşfini mümkün kılmaktadır. YBM, bina enerji performans analizi için gerekli olan verilerin çoğunu içermektedir. YBM uygun şekilde kullanılırsa, enerji simülasyonu oluşturmak için girdi verilerini hazırlarken önemli miktarda zaman ve çabadan tasarruf edilmekte ve hatalar azaltılabilmektedir. YBM tabanlı performans optimizasyonu çerçevesi Şekil 4.39'da yer almaktadır. Kullanılan yöntem tasarımcıların tasarım alternatiflerini görsel bir programlama ara yüzü kullanarak keşfetmelerine olanak tanırken, tasarım modellerinin en uygun alternatifini bulmak için enerji performanslarını değerlendirmektedir. Asl ve diğerleri (2015b) ortaya koydukları bu çalışmada tasarımcının erken tasarım sürecinde çoklu hedefleri optimize etmesini sağlamak için entegre bir sistem geliştirmişlerdir. Bina enerji performansını optimize etme çerçevesinde, Revit'te parametrik YBM modelleri oluşturmak ve GBS'deki (Green Building Studio) parametrik simülasyonları çalıştırmak için temel işlevleri içermek üzere Dynamo düğümleri geliştirilmiştir. Bir MOO algoritması (Non-dominated Sorting Genetic Algorithm-II veya NSGA-II), tasarımcıların birden fazla çelişen hedefi optimize etmesine ve bir dizi optimal çözüme yaklaşmasına yardımcı olabilecek bir düğüm paketi olarak Dynamo'da Optimo paketi ile oluşturulmuştur.

Bir konut binasının uygun günışığı seviyesini maksimuma çıkarırken enerji kullanımını minimuma indiren örnek uygulama, sistemin ve iş akışının faydasını göstermek için kullanılmıştır (Şekil 4.40).



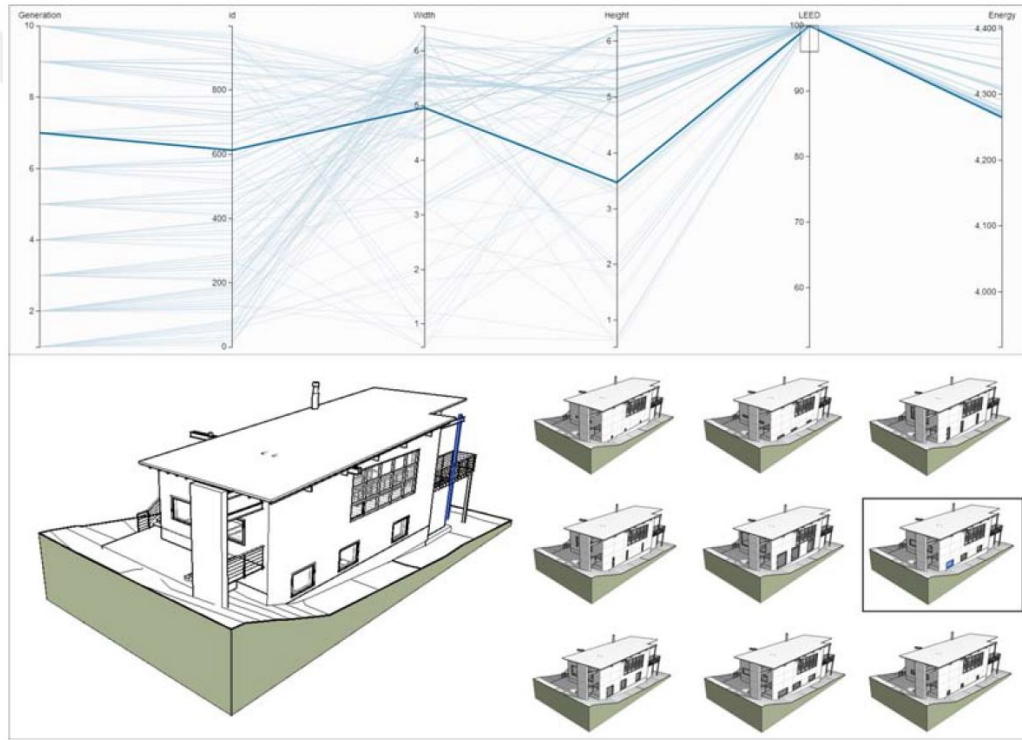
Şekil 4.39 : YBM tabanlı performans optimizasyon çerçevesine genel bakış. (Asl ve diğ., 2015b)



Şekil 4.40 : Optimizasyon süreci iş akışı (Asl ve diğ., 2014)

Bu optimizasyon çalışmasının hedefi, beklenen enerji kullanımını en aza indirirken, konutta yer alan günışığı için gereksinimleri karşılanan oda sayısını en üst düzeye çıkarmaktır. Enerji kullanımının simülasyonu ve hesaplanması, YBM'nin sağlayabileceği yapı bilgisini, örneğin geometri bilgisi, fiziksel materyal bilgisi ve

modele yerleştirilmiş konum verilerini gerektirmektedir. Dynamo ortamında yapı bilgi modeline ait olan optimizasyon sürecine dahil edilecek değişken parametreler tanımlanır. Bu örnek için pencerelerin genişliği ve yüksekliği Dynamo ortamında değişken parametreler olarak tanımlanmıştır. Değişken parametreler binanın yönelimi gibi çok farklı özellikler hakkında da olabilmektedir. Parametreler değiştikçe optimizasyon çıktısı olarak okunan enerji kullanımı değerleri de değişmektedir. Şekil 4.41’de koyu çerçeve ile seçilmiş tasarım alternatifi en yüksek LEED kredili, en düşük enerji kullanımına sahip tasarım alternatifini göstermektedir. Seçilen alternatife ait değerler grafik üzerinden okunabilmekte ve diğer alternatiflerle eş zamanlı karşılaştırma yapılabilmektedir.



Şekil 4.41 : Tasarım alternatiflerine ait sayısal değerlerin ve üç boyutlu modellerin görüntülenmesi (Asl ve diğ., 2014)

Araştırma, simüle edilmiş çok sayıda parametrik tasarım varyasyonu oluşturmak için bir YBM modelinin kullanılmasının, uygun bir iş akışı olduğunu göstermektedir. Bulut tabanlı bir enerji analiz aracının (Green Building Studio) kullanılabilirliği yüzlerce tasarım varyasyonunun hızlı bir şekilde değerlendirilmesini sağlamakta ve görsel programlama ile bağlantı kurulması tasarım alternatiflerinin hızlı ve doğru bir şekilde belirlenmesini sağlamaktadır. YBM içinde gömülü olan veriler, olası tasarım kararlarına, performans etkilerini ölçmek için hızlıca uygulanabilmektedir. Sınırlı

parametrik modelleme ve programlama becerisine sahip tasarımcılar, çok çeşitli tasarım analizleri gerçekleştirmek için Optimo paketi içerisinde yer alan düğümleri kullanabilmektedir. Böylece, mimar ve mühendisler, erken tasarım aşamasında YBM tabanlı enerji optimizasyon yöntemlerini kullanarak projenin karar değişkenlerinin tasarım üzerindeki etkilerini anlayabilmektedir.

Georgia Institute of Technology ve Stanford University gibi bazı üniversitelerin, lisans ve yüksek lisans sınıflarında Optimo ile ilgili dersler verilmeye ve araştırmalarında da uygulanmaya başlamıştır. Ayrıca Optimo, Avrupa'da Arup ve Autodesk Structural Analysis ekibi tarafından sektördeki gerçek tasarım projelerinde de denenmiştir.

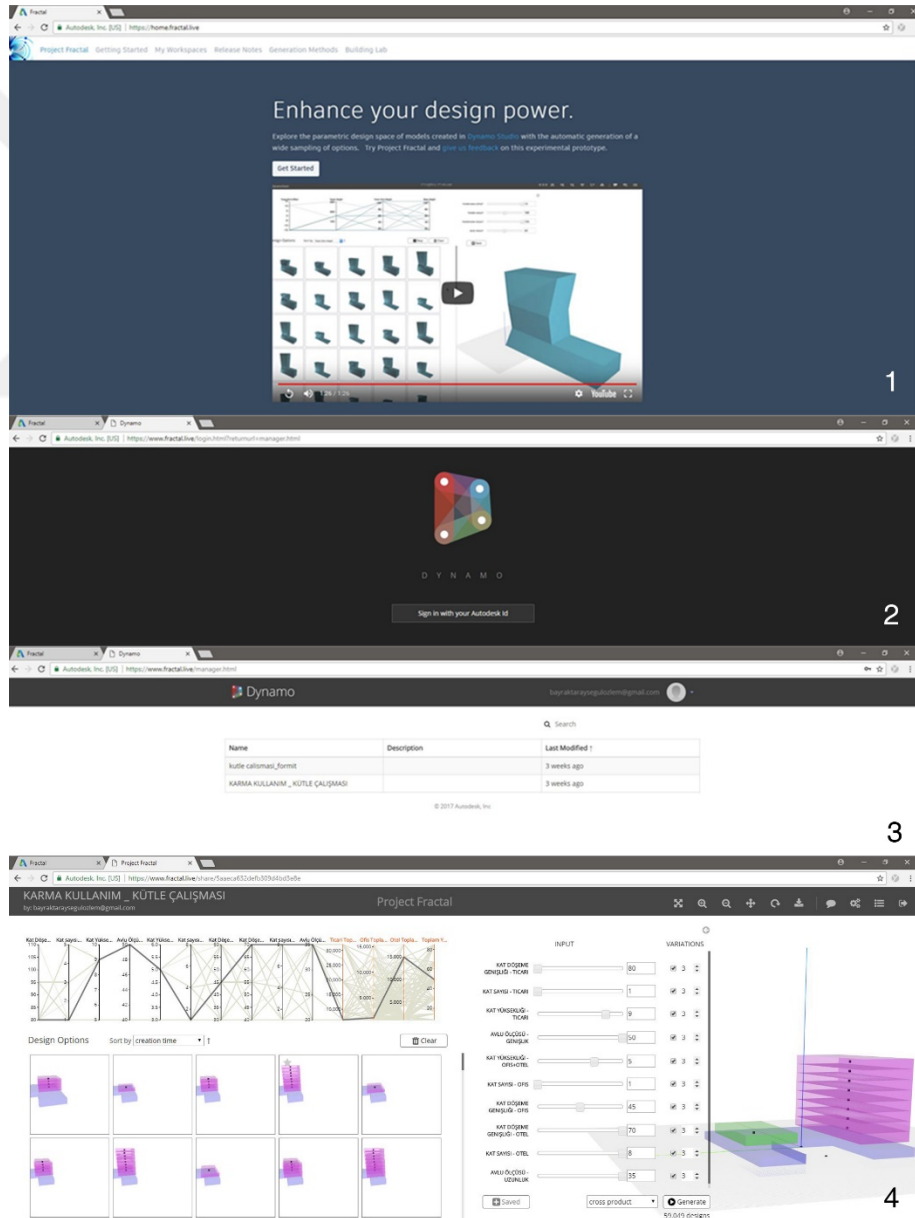
4.2.1.4 Project Fractal ile yapılan uygulamalar

Kasım 2015'te geliştirilmeye başlanan ve Temmuz 2016'da kullanıma uygun hale getirilen Project Fractal, birden fazla tasarım problemine uygulanabilecek açık bir sistem olarak Autodesk tarafından oluşturulmuştur. Tasarım çözümlerinin hızlı bir şekilde değerlendirilmesi ve farklı kombinasyonların oluşturulabilmesi için bulut tabanlı sistem kullanılmaktadır. Başlangıç için Project Fractal'ın internet sitesi (<https://home.fractal.live/>) üzerinden Dynamo Studio çalışma alanına giriş yapılması gerekmektedir. Dynamo Studio ortamında hazırlanan algoritmaların internete yayınlanması ile birlikte Dynamo dosyaları Project Fractal ile kullanılmak üzere hazır hale gelmektedir (Şekil 4.42).

Project Fractal gibi araçlar çeşitli akademik kurumlarda ve profesyonel firmalarda geliştirilip araştırılırken, bulut bilişimin kullanılabilirliğinin artmasıyla birlikte MMİ endüstrisi ve akademi için tasarımının sayısal olarak sunulması hızlandırılmış bir şekilde karşımıza çıkmaktadır. Project Fractal'ın birbirinden farklı kombinasyonlar üretmesi ile birlikte tasarımcının üretilebilecek olan olası çözümler ile çözüm kümesini genişleterek, daha önce zaman ve maliyet kısıtlamaları nedeniyle üretilmeyen opsiyonlar ile ilgili daha fazla keşif yapılması sağlanmaktadır. Üretken ve analitik teknikler kullanılarak, tasarımcılar için kritik olarak alınması gereken kararlar, çok çeşitli alternatifler denenerek hızlıca alınabilmektedir.

Project Fractal ile oluşturulması hedeflenen tasarım alternatifleri için bir dizi önemli hedef, gereksinim ve kısıtlamaların belirlenmesi gerekmektedir. Dynamo ortamında ilgili düğümlerin “girdi” olarak işaretlenmesi durumunda sayısal Dynamo kaydırıcıları

veya doğrudan sayısal değer girişleri, Project Fractal ara yüzünde görüntülenmekte ve bu parametreler tasarım alternatifleri üretmek için Project Fractal üzerinden değiştirilebilmektedir. Proje girdilerine bağlı olarak, Project Fractal'ın arabiriminde bazı çıktı değerlerinin görüntülenmesi istenildiğinde, Dynamo ortamında istenilen parametreye bir izleme düğümü (watch node) eklenmesi gerekmektedir. Bu eklenen düğüm ile proje çıktılarının sayısal ve grafik değerleri, Project Fractal ara yüzünde görüntülenmekte ve bu çıktı değerlerine göre filtreleme yapılabilir. Projeye ait ilgili girdi ve çıktı parametrelerinin ayarlanmasından sonra Dynamo'da yer alan "Web'e Gönder" seçeneği kullanılarak ilgili algoritma Project Fractal ortamına yüklenmektedir.



Şekil 4.42 : Project Fractal ve Dynamo Studio'nun birlikte kullanılması

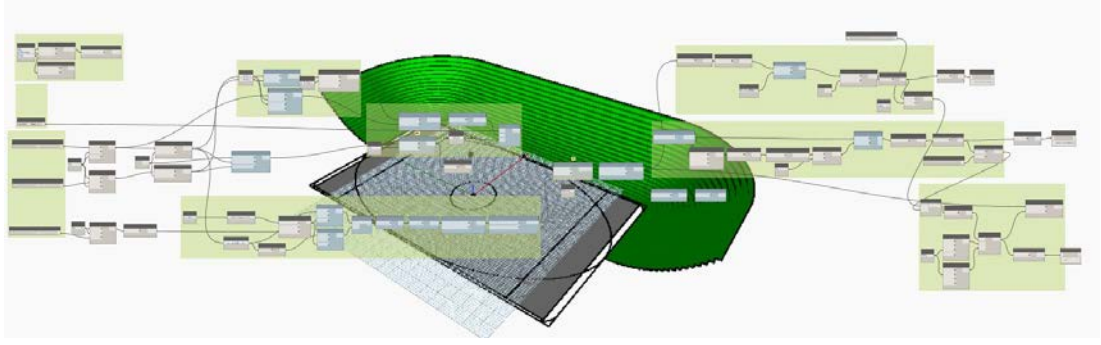
Project Fractal, her bir girdinin maksimum, minimum ve ortalamasını, tasarım alternatifleri oluşturmak için herhangi bir sayısal değerle birleştirir. Grafik farklı giriş değerleriyle çalıştıkça farklı kombinasyonlar ortaya çıkmaktadır. Oluşturulan bu tasarım alternatifleri küçük resim ön izlemeleri şeklinde ara yüzde görüntülenmektedir. Oluşturulan herhangi bir tasarım alternatifinin üzerine gelindiğinde sayfanın sol üst köşesindeki paralel koordinat grafiğindeki koyu renkli olarak beliren çizgiler ve değerler seçilen tasarım alternatifine ait giriş ve çıkış değerlerini vurgulamaktadır. Paralel grafik interaktif bir haldedir. Filtre sınırını oluşturmak için oluşturulan tasarımlara ait olan bir giriş veya çıkış sütununa tıklayarak ve sürükleyerek filtreleme yapılabilmektedir. Filtre sınırı, yatay sınırlarını tıklatıp sürükleyerek veya tüm sınırı dikey eksen boyunca sürükleyerek ayarlanabilmektedir. Ayrıca, sıralama menüsünü kullanarak tek bir giriş veya çıkış değerine göre, beğenilen tasarımlara göre alternatifleri sıralamak mümkündür. En uygun tasarım alternatifi tasarımcı tarafından manuel bir şekilde seçilmektedir. Project Fractal'da yer alan indirme seçeneği, Dynamo (.dyn) dosyasını, .stl formatında görüntülenen tasarımın geometrisini (bu format Formit'e dosya aktarımı sırasında kullanılmaktadır) veya görüntülenen tasarımın .csv biçimindeki tanımlı giriş ve çıkış değerlerini indirmek için (Excel gibi tablolar) belirli opsiyonlar sunmaktadır (Şekil 4.47). Project Fractal ile yapılan uygulamalara iki örnek verilmiştir:

1. Autodesk Dynamo ekibi tarafından hazırlanan stadyum tasarımının alternatiflerinin Project Fractal ortamında oluşturulması (Autodesk Project Fractal, t.y.)
2. Project Fractal ile karma kullanım fonksiyonlu bir yapı bloğu için form alternatiflerinin oluşturulması

Bu uygulamalar sırası ile detaylı bir şekilde ele alınmaktadır.

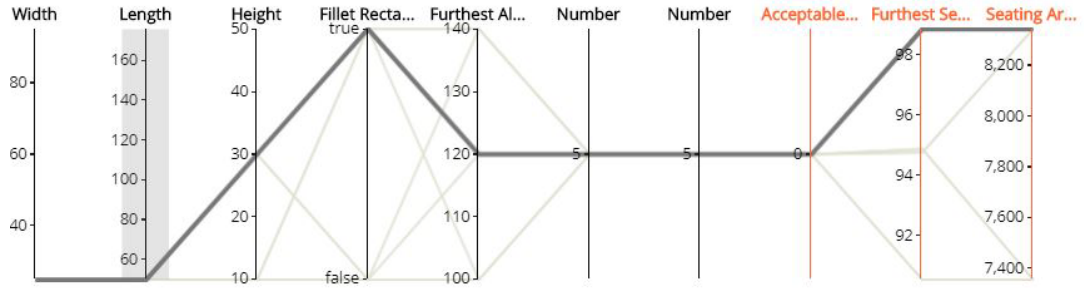
Autodesk Dynamo ekibi tarafından hazırlanan stadyum tasarımının alternatiflerinin Project Fractal ortamında oluşturulması

Project Fractal ile Dynamo'da stadyum oturma alanı için hazırlanan algoritma (Şekil 4.43) üzerinden parametrelerin tanıtılması ile farklı tasarım alternatifleri oluşturulabilmektedir.

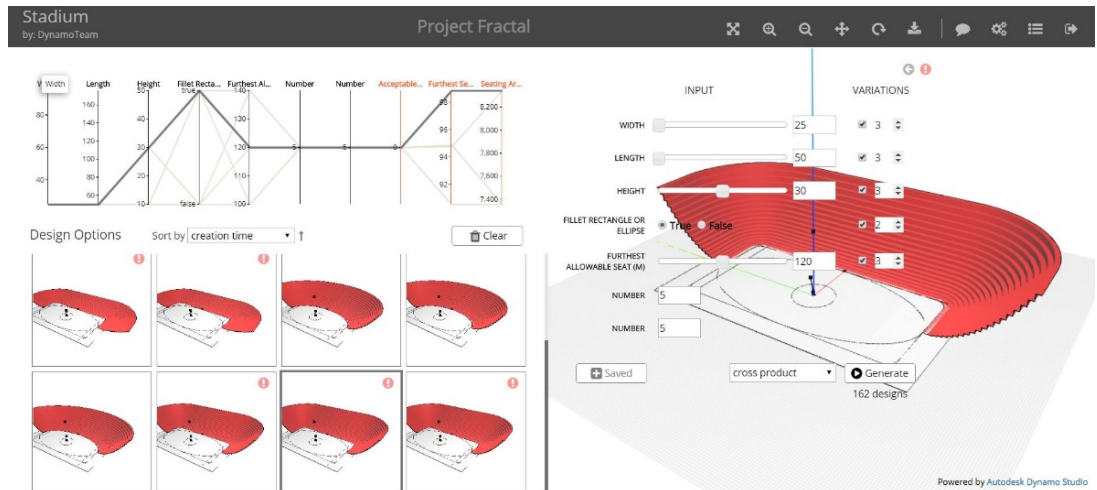


Şekil 4.43 : Autodesk Dynamo Studio ile oluşturulan stadyum algoritması ve ön izlemesi (Autodesk Project Fractal, t.y.)

Dynamo ortamında parametrize edilen tasarım girdileri Project Fractal’da yer alan değişkenleri temsil etmektedir. Project Fractal değişkenler grafiğinin (Şekil 4.44) sol tarafında yer alan değişkenler tasarım ile ilgili girdileri, sağ tarafında turuncu renk ile gösterilen değişkenler ise proje çıktılarını temsil etmektedir. Grafikte değişken parametreler yan yana sıralanırken değişkenlerin isminin altında ise sayısal değerleri yer almaktadır. Bu değişkenlerin görsel karşılığı da aynı ara yüz içerisinde kullanıcıya sunulmaktadır (Şekil 4.45).

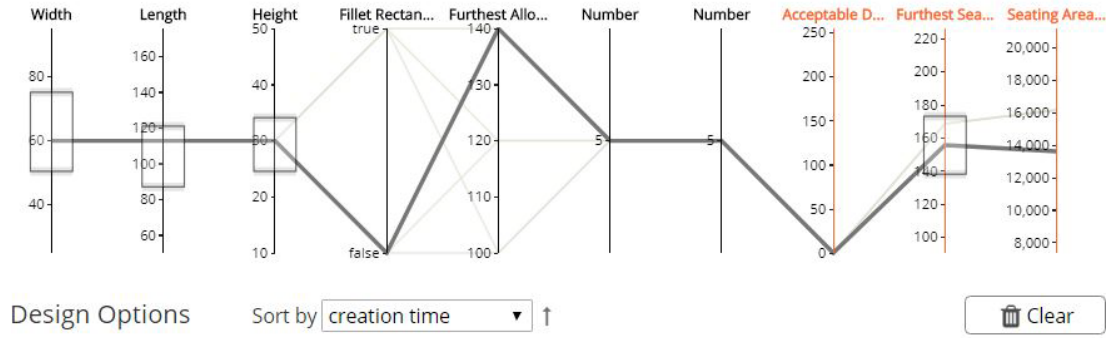


Şekil 4.44 : Project Fractal değişkenler grafiği (Autodesk Project Fractal, t.y.)



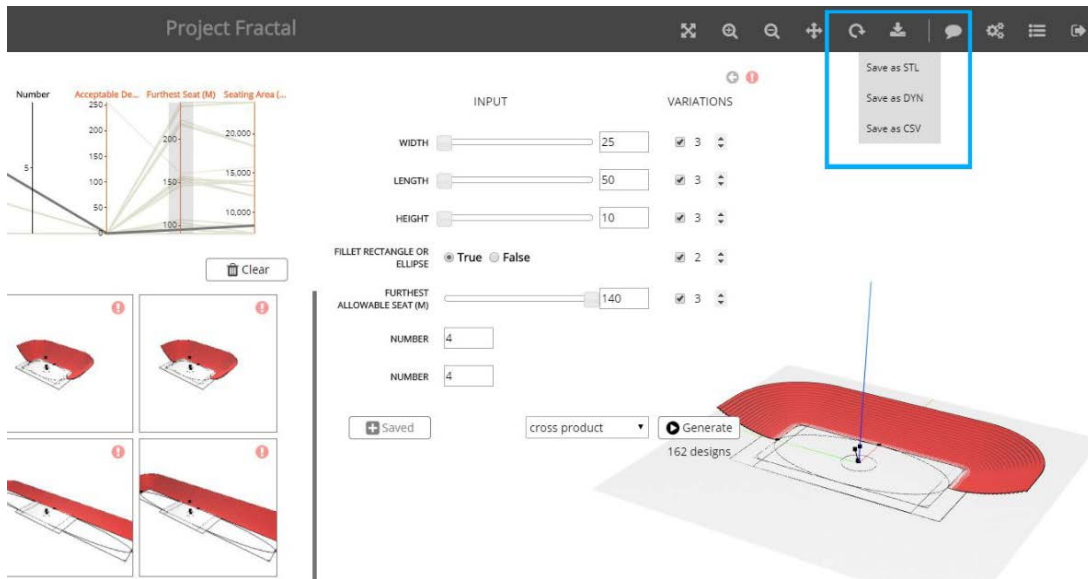
Şekil 4.45 : Project Fractal ortamında oluşturulmuş olan tasarım alternatifleri (Autodesk Project Fractal, t.y)

Oluşturulan tasarım alternatifleri arasından seçilen alternatif daha koyu bir çerçeve ile belirtilmektedir. Ara yüzün sağ tarafında yer alan görselde ise seçilen alternatifin büyütülmüş hali ile birlikte değişkenlerin sayısal karşılığı gösterilmektedir. Ayrıca sol üstte yer alan değişkenler grafiği üzerinden de seçili alternatifin değerleri ile diğer alternatiflerin değerleri birbirleri ile karşılaştırılarak ayırt edilmektedir (Şekil 4.45). Ayrıca tasarımcılar optimum tasarım alternatiflerine ulaşabilmek için değişkenler grafiği üzerinden filtrelemeler yapabilmektedir (Şekil 4.46).



Şekil 4.46 : Project Fractal değişkenlerinin filtrenmesi (Autodesk Project Fractal, t.y.)

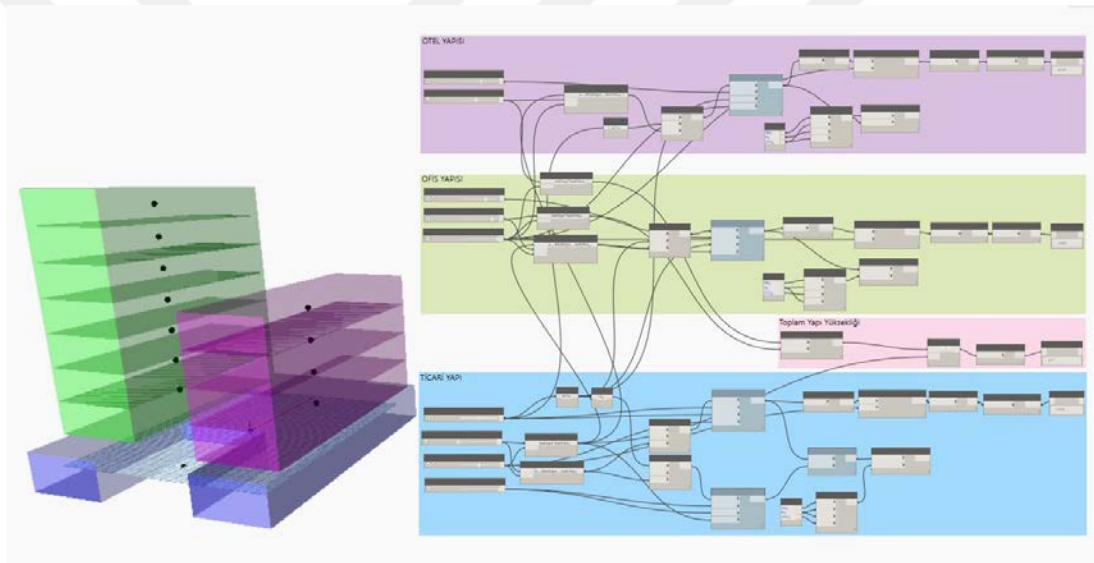
Tasarım alternatifleri üretilme sürecinde elde edilmesi hedeflenen alternatife ulaşıldığı zaman bu süreç durdurularak, sonuç tasarım alternatifi diğer yazılım ortamlarına aktarılabilmektedir (Şekil 4.47).



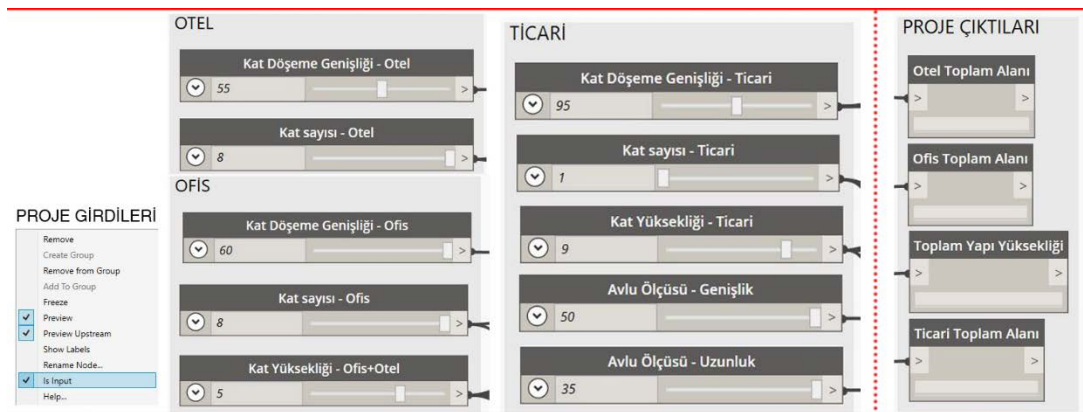
Şekil 4.47 : Project Fractal ortamından tasarımı dışarıya aktarma seçenekleri (Autodesk Project Fractal, t.y.)

Project Fractal ile karma kullanım fonksiyonlu bir yapı bloğu için form alternatiflerinin oluşturulması

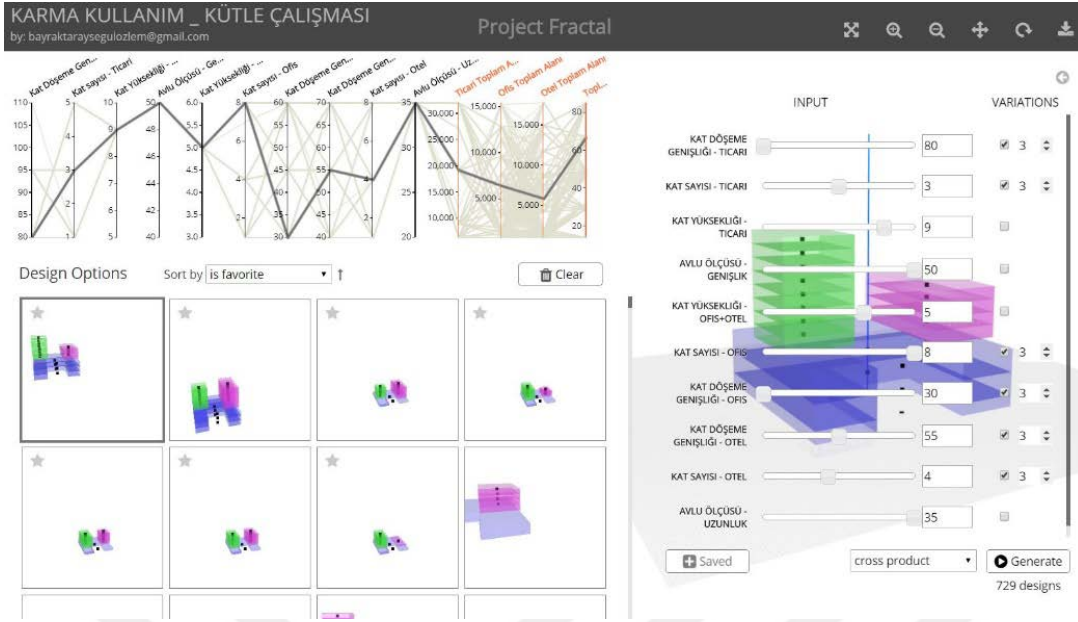
Erken tasarım aşamasında karma kullanımlı bir projeye ait kütle çalışması, Dynamo Studio ortamında oluşturulmuştur (Şekil 4.48). Oluşturulan kütleyle ait parametreler Dynamo’da girdi ve çıktı olarak (Şekil 4.49) tanımlandıktan sonra Project Fractal ile farklı tasarım alternatifleri oluşturulmuştur (Şekil 4.50). Project Fractal ortamında oluşturulan alternatiflerin filtrelenmesi ve Dynamo ortamında tanımlanan değerlere göre oluşan tasarım alternatifleri arasından sonuç ürün tasarımcılar tarafından seçilmektedir. Şekil 4.50’de seçilen alternatif koyu çerçeve ile gösterilmekte, ilgili parametrelerin sayısal değerleri değişkenler grafiğinde belirgin bir hale gelmekte ve sağ taraftaki alanda alternatifin ön izlemesi görüntülenmektedir.



Şekil 4.48 : Dynamo’da oluşturulan karma kullanımlı yapı örneği algoritması ve ön izlemesi



Şekil 4.49 : Dynamo’da oluşturulan algoritmaya ait girdi ve çıktı parametreleri

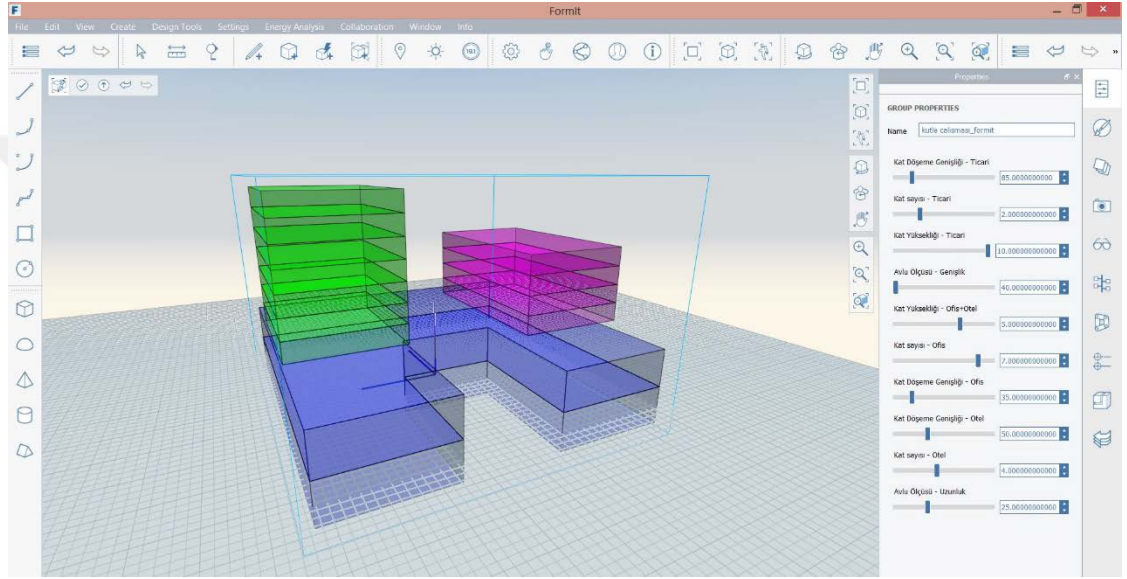


Şekil 4.50 : Project Fractal ile oluşturulan tasarım alternatifleri

Project Fractal, Dynamo ile birlikte kullanıldığında, MMİ endüstrisinde optimizasyonu teşvik etme potansiyeline sahip üretken bir tasarım ortamı oluşturmaktadır. Hem MMİ uzmanları hem de ticari yazılım şirketleri tarafından yazılan web servisleri tarafından sağlanan uygulanabilir bilgi ile Project Fractal ve Dynamo ile ortaya konulan yöntem, proje süreçleri için faydalı bir şekilde kullanılabilir. Bina işlev tanımları, geometrik temsiller, alan planlama tabloları ve şemaları, performans gereksinimleri ve kısıtlamaları ile oluşturulan tasarım hedeflerini elde edebilmek için oluşturulması gereken alternatifler Project Fractal ile birlikte oluşturulurken zaman ve maliyetten tasarruf edilmektedir.

Projenin erken tasarım aşamasında Dynamo Studio ile hazırlanan algoritma Formit programına aktarılabilmektedir. Formit ile Dynamo birlikte çalışabilmektedir. Formit ortamında hazırlanan arsa ve yakın çevre ile ilgili verilerle birlikte tasarım problemi olarak ele alınan tasarım birleştirilebilmektedir. Formit ortamında Dynamo'da tanımlanan girdiler değiştirebilmektedir. Böylece erken tasarım aşamasında YBM tabanlı kavramsal tasarım aracı olan Formit ile Dynamo tek bir arayüz üzerinden eş zamanlı olarak çalışabilmektedir. Projenin her aşamasında YBM tabanlı sistemlerin kullanılması ve parametrelerden erken tasarım aşamasından itibaren söz edilmesi birçok tasarım alternatifini oluştururken zaman kaybını önlemektedir. Formit aracı ile elde edilen tasarım, yapı bilgi modelleme platformu olan Revit gibi programlara veri kaybı olmadan aktarılabilmektedir. Böylece proje yaşam döngüsünün ilk evresinden

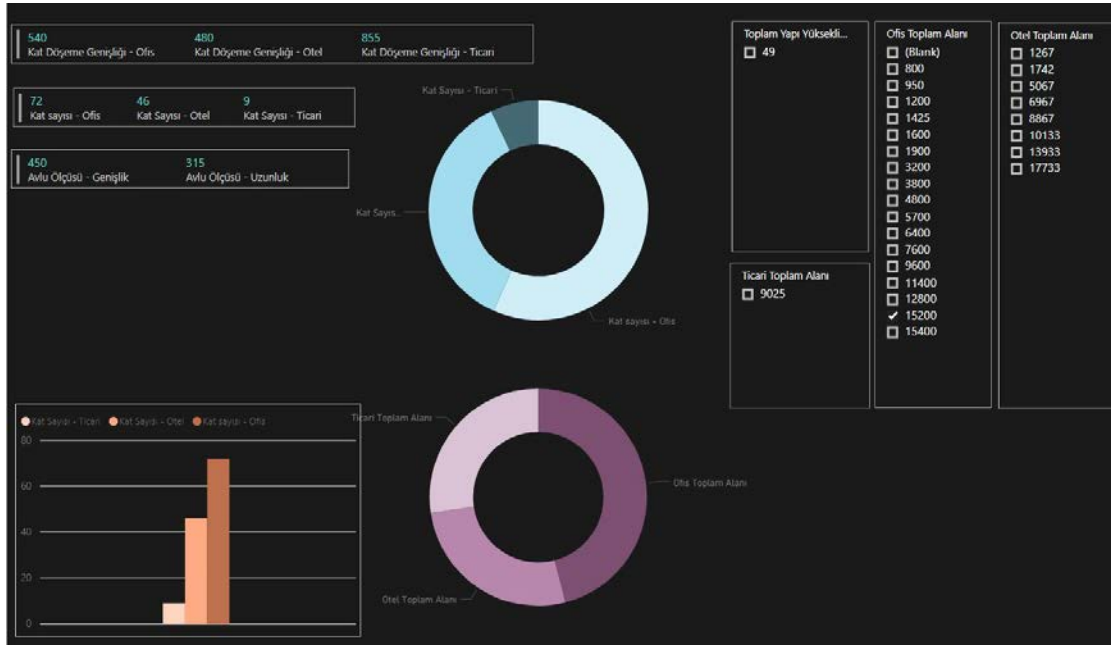
detaylandırma evresine kadar olan süreç de yapı bilgi modellemesi ile birlikte hesaplamalı tasarım ve kavramsal tasarım araçları veri kaybı olmadan birlikte kullanılmış olmaktadır. Formit ve Dynamo'nun kavramsal tasarım aşamasında birlikte kullanılmasına, karma kullanımlı proje için hazırlanmış olan algoritma üzerinden örnek verilebilir (Şekil 4.51). Dynamo ortamında hazırlanan algoritmada tanımlanan girdiler Formit ortamında görüntülenebilmektedir. Girdi olarak tanımlanan parametrelerin değiştirilmesiyle birlikte Formit ortamında yapılan değişiklikler eş zamanlı olarak görüntülenebilmektedir.



Şekil 4.51 : Formit ortamında Dynamo parametrelerinin değiştirilmesi

Project Fractal ortamında elde edilen tasarım alternatifleri .csv formatında tablo halinde dışarıya aktarılabilir. Bu formatta tasarım alternatiflerine ait veriler Excel gibi programlar aracılığı ile görüntülenebilmektedir. Tasarım opsiyonlarına ait veriler Power BI gibi data görselleştirme amacıyla kullanılan programlar aracılığı ile görselleştirilerek kullanıcının ihtiyaçlarının ne ölçüde karşılandığı kolayca analiz edilebilmektedir (Şekil 4.52). Power BI ortamına aktarılan farklı tasarım alternatiflerine ait veriler eş zamanlı olarak birbirleri ile kıyaslanabilmektedir. Tasarım alternatiflerine ait elde edilen bütün verilerin tek bir ara yüzde kullanıcıya sunulması karar verme aşamasında kullanıcıya yardımcı olmaktadır.

Project Fractal'in Dynamo Studio, Format ve Power BI gibi programlarla birlikte kullanılması tasarımcıya yeni olanaklar açmaktadır. Oluşturulan bu çerçeve kapsamında kullanılabilecek olan programlar tasarımcının ihtiyaçları doğrultusunda artırılabilir.

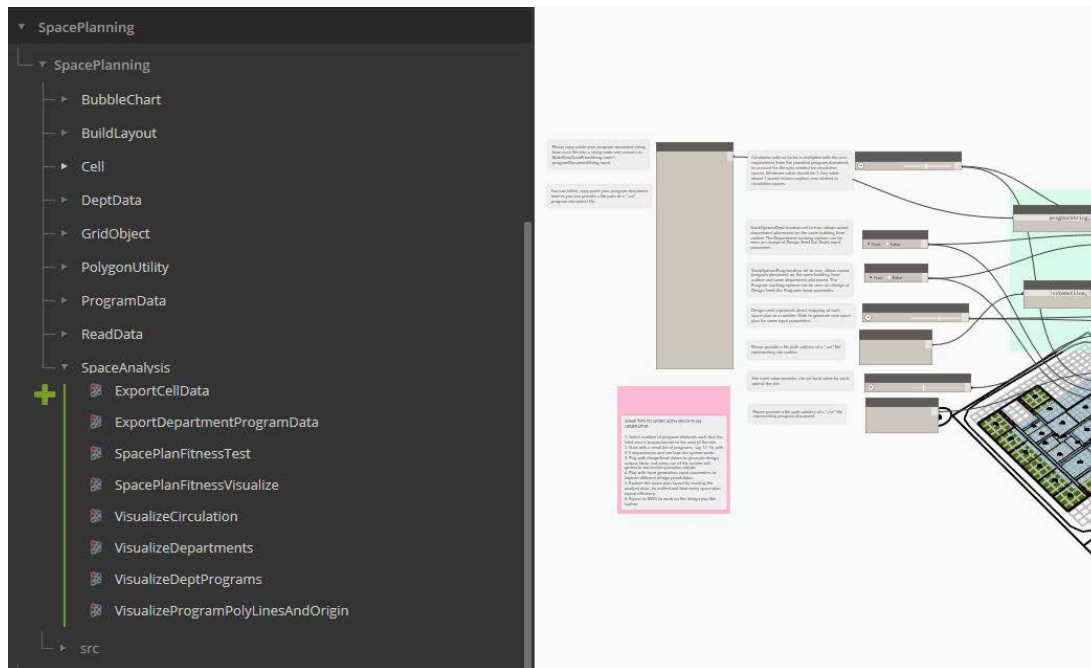


Şekil 4.52 : Tasarım alternatiflerine ait verilerin Power BI ortamında görselleştirilmesi

Power BI ortamında görselleştirilen verilere ait diyagramlar üzerinden seçim yapılarak sonuçlar eş zamanlı olarak görüntülenebilmektedir.

4.2.1.5 Mekân planlama ve optimizasyonu (Space Planning)

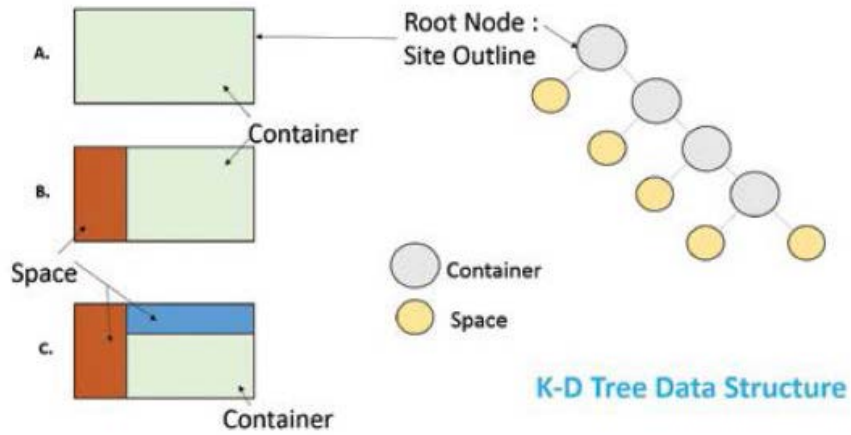
Mekân Planlama Türeticisi (Space Planning), Dynamo paketi olarak geliştirilmiştir. Paketin içerisinde ilgili birçok düğüm bulunmaktadır (Şekil 4.53).



Şekil 4.53 : Dynamo Space Planning paketi

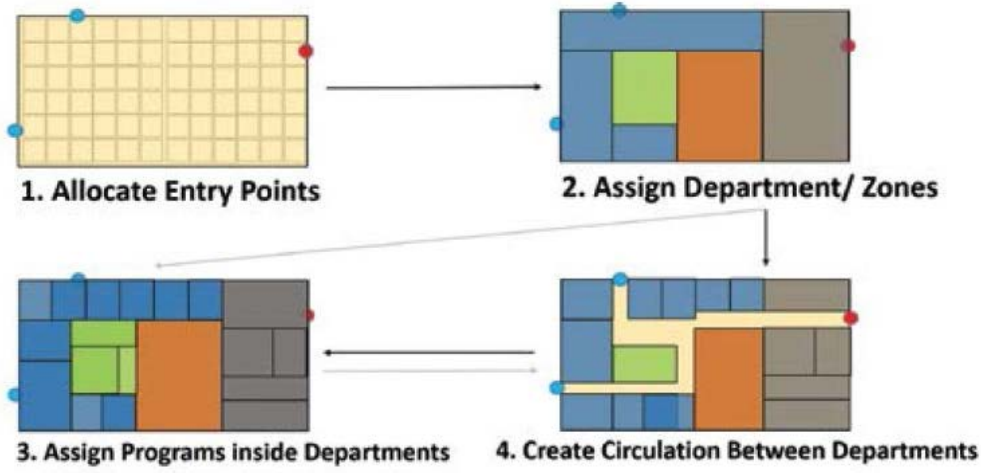
Mekân Planlama Türeticisi, mimari mekân planlarının hızlı üretimi ve analizini mümkün kılan otomatik bir hesaplama yöntemi geliştirmek için geliştirilmiştir. Mekân Planlama eklentisi ile önerilen iş akışının tasarım sürecinin erken aşamalarında proje hedeflerini netleştirebileceği, zamandan tasarruf sağlayacağı, daha iyi kaynak tahsisine olanak sağlayabileceği, diğer disiplinlerden olan paydaşlarla birlikte daha bilinçli kararlar verilebileceği ve daha iyi tasarımlar sunulmasına yardımcı olabileceği düşünülmektedir. Bu kapsamda mekân planlama eklentisi bir hastane projesinin tasarımı ile birlikte test edilmiştir (Das ve diğ., 2016).

Mimari tasarım, tasarım hedeflerine ulaşılan kadar çok sayıda tekrarlama gerektiren bir süreçtir. Bu nedenle mekân planlama türeticisi (space plan generator) aynı verilere birden çok kez erişebilen ve mekânsal verileri en yakın komşu mahallere göre arayabilecek şekilde organize edilmiş bir veri yapısı kullanmaktadır. Mekânsal bölümlenme yapısına uygun olarak K-d veri strüktürü bu amaç doğrultusunda kullanılmaktadır. İç içe geçmiş öğeler, çift yönlü geçiş, yukarıdan aşağıya ve aşağıdan yukarıya doğru verileri almak için kullanılmaktadır (Şekil 4.54).



Şekil 4.54 : Mahal, departman ilişkisi ve K-D Veri Strüktürü (Das ve diğ., 2016)

Mekân planlama türeticisinin yaklaşımı hiyerarşiktir. Program öğeleri (mahal, departman, sirkülasyon vs.) proje alanına eklenecek elemanlardır. Özyinelemeli algoritma önce departmanı, daha sonra onların içindeki mahalleri yerleştirir ve son olarak departmanlar arasındaki sirkülasyonu oluşturur (Şekil 4.55). Bazı program unsurları bölümleri oluşturmak için kümelenebilir. Burada departmanlar ana küme olarak çalışır. Mahaller ise departmanların bir alt kümesi olarak çalışmaktadır (Das ve diğ., 2016). Bu durum Şekil 4.54’de yer alan K-d veri strüktürü çalışma prensibinde belirtilmiştir.



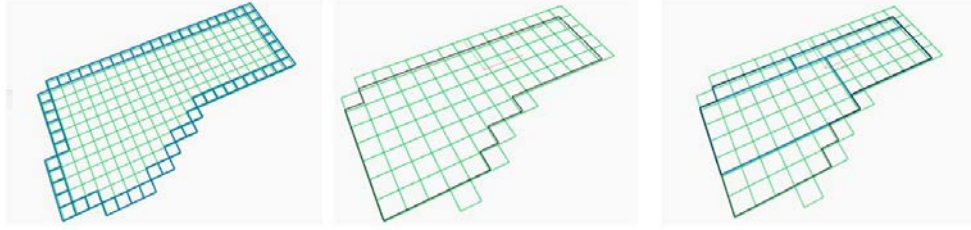
Şekil 4.55 : Hiyerarşik mekân planlama yaklaşımı (Das ve diğ., 2016)

Bu yaklaşıma göre öncelikle projeye ait giriş noktaları belirlenir. Bu giriş noktalarına göre departmanlar önem sırasına göre yerleştirilir. Oluşturulacak olan mekanlar ile ilgili komşuluk durumları, önem sıralaması gibi bilgilerin bir CSV tablosu üzerinden programa tanıtılması gerekmektedir (Şekil 4.56). Bu tablo üzerinden programa, projede gerekli olan mahallerin adı, adetleri ve alan değerleri tanıtılmaktadır. Ayrıca hangi mahallerin ana mahal olacağı belirtilirken hangi mahallerin birbiriyle komşu olacağı da tanımlanmaktadır.

ID	PROGRAM NAME	ZONE/ DEPARTMENT	QUANTITY	AREA	TOTAL AREA	PREFERENCE	ADJACENT PROGRAM
0	Family Lounge	Family Public Support	48	15	720	3	3..4..5..8
1	Public Toilet	Family Public Support	2	160	320	1	2..4..6..15
2	Conference	Family Public Support	1	225	225	2	3..2
3	Patient Room	Inpatient Area	26	340	8840	8	12..11..5..6..8
4	Patient Room Isolation	Inpatient Area	2	340	680	9	2..3..5..7..8
5	Patient Room VIP	Inpatient Area	2	340	680	10	1..3..2..4..5
6	Alcove Charting Room	Inpatient Area	16	35	560	5	0..4..6
7	Team Work station	Clinic Support	2	288	576	8	7..4..3..5..8..9
8	HUC Station	Clinic Support	1	371	371	10	2..4..6
9	Off stage Work Area	Clinic Support	2	323	646	3	9..15..3
10	Crash Cart Alcove	Clinic Support	3	30	90	4	11..10..14..5
11	Equipment Storage	Clinic Support	2	320	640	2	4..6..7..9..11..13
12	Clean Utility/ Supplies	Clinic Support	2	200	400	6	2..3..5..7..8
13	Soiled Utility	Clinic Support	1	300	300	7	1..3..2..4..5
14	Chute Room	Clinic Support	1	150	150	1	0..4..6
15	EVS Closet	Clinic Support	1	80	80	0	7..4..3..5..8..9
16	Nourishment Room Patient	Clinic Support	1	120	120	2	2..4..6
17	Nourishment Room Family	Clinic Support	1	40	40	2	2..4..6..15
18	Medication Room	Clinic Support	3	150	450	8	3..2
19	Alcove Portable Med Equip	Clinic Support	2	40	80	3	12..11..5..6..8
20	Unit Specific Storage Space	Clinic Support	1	300	300	4	2..3..5..7..8
21	Respiratory Therapy Workroom	Clinic Support	1	120	120	8	1..3..2..4..5

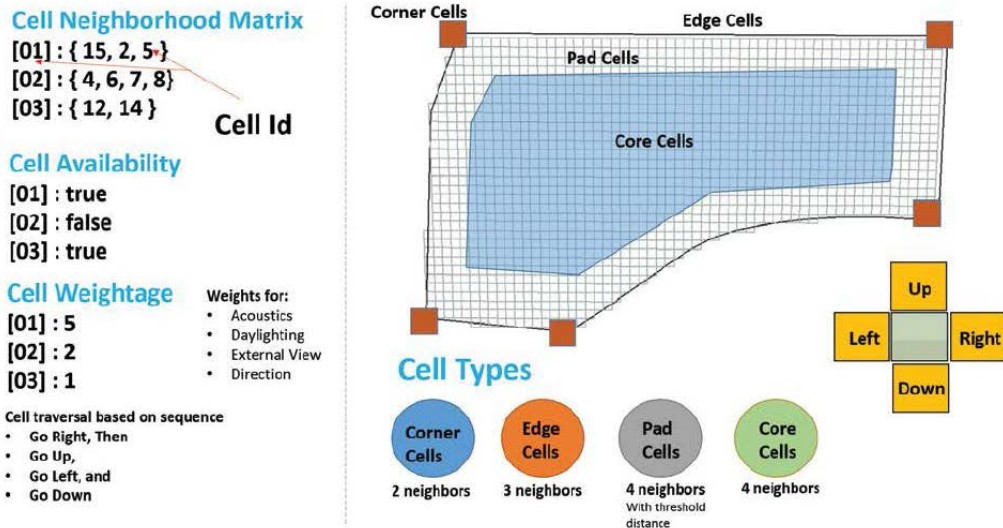
Şekil 4.56 : Mekân Planlama Türeticisi için oluşturulan CSV tablosu (Das ve diğ., 2016)

Proje alanının sınırları bir SAT dosyasıyla Dynamo’da hazırlanan algoritma içerisinde tanımlanmaktadır. Tanımlanan proje alanının sınırlarının belirli mesafede ötelenmesiyle yapı için sınırlayıcı bir alan oluşturulmaktadır (Şekil 4.57).

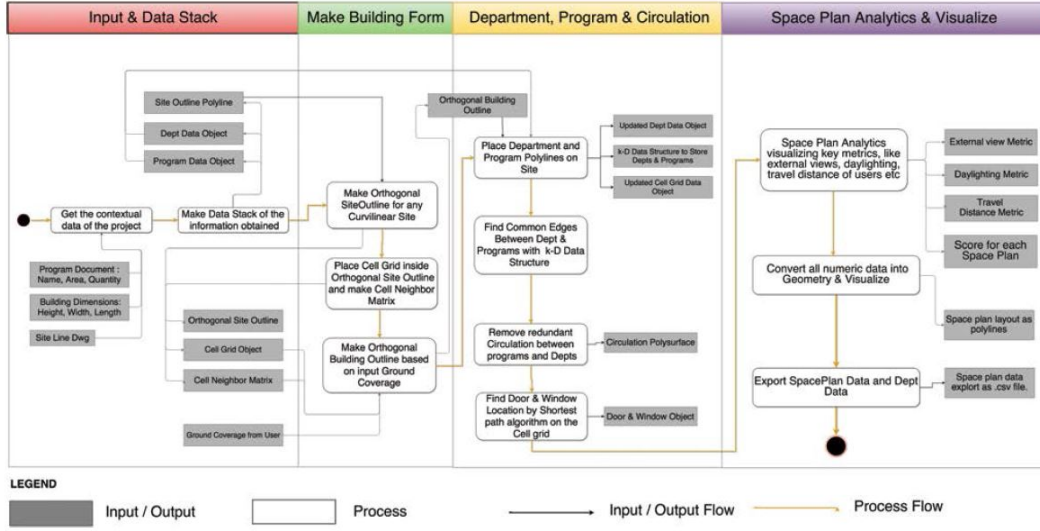


Şekil 4.57 : Proje sınırlarının ve alt bölümlenmelerin oluşturulması (Das ve diğ., 2016)

Algoritma, sınırlanan alan üzerinde, kullanıcı tarafından belirtilen aralıklarla oluşturulan bir ızgara üzerinde noktaların oluşturulmasını sağlamaktadır. Izgara nesneleri, her hücrenin komşularını üç boyutta izleyen hücre komşu matrisini oluşturmak için kullanılır. Böyle bir matrisin yararı, alanı verimli bir şekilde çaprazlama kabiliyeti sunmasıdır. Izgara nesnesindeki her hücre bir konum kimliğine sahiptir. Bir hücrenin komşu matrisi, tüm komşu hücrelerin tanımlayıcılarını saklayan bir listeye sahiptir. Komşu hücreler, yukarı, sağa, sola ve aşağı doğru sırayla hareket eder. 0 olarak tanımlanan bir hücre, komşu matris dizisinde saklanan 05, 08, 02, 15 hücrelerine sahip olabilir, 05 sağ taraftaki hücreyi tanımlayabilirken, 08 üst taraftaki hücreyi, 02 sol taraftaki hücreyi, 15 ise aşağıdaki hücreyi tanımlayabilir. -1 herhangi bir komşu olmayan bir hücreyi gösterir. Matris, kapılar ve çıkış yolları gibi belirli yerleri bulmak için alanı bir köşeden diğerine hareket ettirmek için bir araç sağlamaktadır (Şekil 4.58).



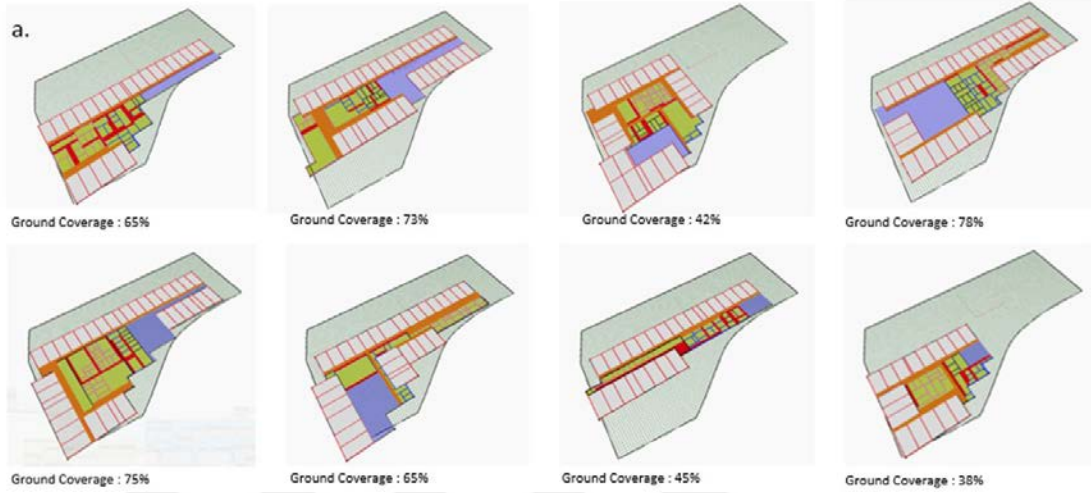
Şekil 4.58 : Mekân planlama türeticisi hücre çalışma mantığı (Das ve diğ., 2016)



Şekil 4.59 : Mekân planlama türeticisi tasarım süreci (Das ve diğ., 2016)

Şekil 4.59'daki grafikte anlatılan Mekân Planlama Türeticisinin (Space Planning) çalışma metodolojisini özetlersek; öncelikle Mekân Planlama Türeticisinin csv dosyası biçiminde proje gereksinimlerinin tanımlandığı bir tabloya ihtiyacı vardır. Her mahall için tercih edilen ve csv dosyası ile tanımlanan değer, mahallerin alana yerleştirileceği sırayı belirtmektedir. Mahal komşu listesi, mahaller alana tahsis edilirken kullanılan bir karşılaştırma çizelgesidir. Csv dosyasında tanımlanabilecek diğer kullanıcı girişleri koridor genişliği, dolaşım alanı ve mahallerin en boy oranı ve ızgara hücre boyutu gibi özellikler olabilmektedir. Csv tablosu ile aktarılan veriler, belirtilen tercih değerlerine göre departman ve mahallerin alana atamasının sırasıyla oluşmasına yardımcı olmaktadır. Hücre ızgarasını hücre komşusu matrisi yoluyla çaprazlayan algoritma ile de departmanların ve mahallerin bölümlenerek oluşması sağlanmaktadır. Bu yinelemeli süreç, hangi form yapım stratejilerinin benimsenmiş olduğuna, tasarım hedeflerinin ve kısıtlamalarının ne kadar iyi karşılandığına bağlı olarak, planlar üretmektedir (Şekil 4.60). Departman ve mahal bölümlenmeleri oluşturulduktan sonra sirkülasyon alanı oluşturulur. Dolaşım hesaplaması, mahaller arasındaki dolaşım ağlarını keşfeder ve daha sonra K-d veri yapısını kullanarak mahaller arasındaki dolaşım ağlarını oluşturur. Mahaller arasındaki en kısa sirkülasyonu oluşturabilmek için hücre komşu matrisi yardımıyla yol bulma algoritmaları kullanılır ve oluşturulan yol boyunca gerekli kapılar ve pencereler program tarafından yerleştirilir. Oluşturulan farklı plan çözümlemeleri içerisinde her bir hücrenin kendisine tahsis edilen mahal veya departman hakkındaki bilgisi depoladığı hücreler topluluğu tarafından saklanmaktadır. Hücreler, kapılara ve pencerelere olan mesafe, dolaşım alanlarından

mahalin görünülebilirliği gibi mekânsal plan ile ilgili analitik bilgileri depolamakta ve bu bilgiler, bir mekânsal plan çözümünün genel başarı derecesini belirlemeye yardımcı olmaktadır. Böylece yapılan puanlama, kullanıcıya, tasarım hedeflerine ve kısıtlamalara göre tasarım alternatiflerinin başarısını tasarımcıya iletebilmektedir.



Şekil 4.60 : Mekân Planlama Türeticisi ile oluşturulan plan alternatifleri (Das ve diğ., 2016)

Sonuç olarak, üretken tasarım stratejileri kullanarak mekânsal planlama çalışmaları ilk kez Dynamo ortamında oluşturulmamıştır. Fakat böyle bir üretken tasarım stratejisini mevcutta bulunan hesaplamalı YBM aracı olan Dynamo ortamında geliştirmek çok sayıda kullanıcının sistemden faydalanmasını sağlayarak erken tasarım aşamasında kısa sürede tasarıma ait birçok plan alternatiflerinin gözden geçirilerek daha bilinçli kararlar verilmesine yardımcı olmaktadır. Planlamanın birçok kriter barındırdığı projelerde mekânsal planlamanın önemi oldukça fazladır. Uygulama örneği olarak ele alınan hastane projesi, yer kısıtlamaları ve kat başına hasta yataklarının maksimize edilmesi gibi belirli müşteri hedefleri ile hemşire seyahat mesafesinin en aza indirilmesi gibi sağlanması gereken hedeflere yönelik oluşturulması gereken planların üretken tasarım stratejileri kullanılarak nasıl otomatikleştirilebileceğini anlamak için önemlidir (Das ve diğ., 2016).

Mekân Planlama Türeticisi aynı zamanda bir önceki başlıkta ele alınan Project Fractal ile birlikte de kullanılabilir. Bu sayede belirlenen kriterlerle Project Fractal web ara yüzü üzerinden denemeler yapılabilmektedir. Proje çıktılarının görüldüğü grafik üzerinden istenilen düzeye yakın optimal sonuçlara ulaşmak için filtreleme yapılarak kullanıcı sonuç tasarım alternatifini seçebilmektedir.

4.2.2 Model yönetimi ve kalite kontrol uygulamaları

Geleneksel Yapı Bilgi Modelleme sisteminin kullanıldığı proje adımlarında proje katılımcılarının yoğun olarak çalışması ve çok zaman harcaması gereken işler katılımcıları yeni iş adımlarının oluşturulmasına yönlendirmektedir. Projelerdeki veri doğruluğunun sağlanması projenin detay seviyesinin artmasıyla birlikte zorlaşmaktadır. Farklı disiplinlerin kendi disiplinine ait bilgileri modele eklemesiyle birlikte modeldeki veri yükü artmaktadır. Veri yükü artan bir modelde veri doğruluğunun sağlanması, ihtiyaç olan verilen modelden alınması ve ilgili disiplinlerle paylaşılması da gittikçe zorlaşmaktadır. Bu bağlamda günümüz MMİ sektörü katılımcıları, projenin farklı aşamalarında veri doğruluğunu sağlamak, tekrar eden iş yükünü ve bu işler için harcanan zamanı azaltmak nedeniyle hesaplamalı YBM aracı olan Dynamo'nun olanaklarından nasıl yararlanabileceğinin yollarını araştırmaktadır.

Bu başlık kapsamında ön tasarım aşamasından proje teslim aşamasına kadar olan süreçte hesaplamalı YBM aracı olan Dynamo'nun, proje süreçlerindeki veri doğruluğunu sağlarken veri paylaşılması sırasındaki kayıpları nasıl en aza indirdiğinin yanı sıra belirli işleri otomatize ederek iş yükünden ve zamandan nasıl tasarruf edilebileceği araştırılmaktadır.

Hesaplamalı tasarım, geometrik temsil ile ilgilenmenin yanı sıra yapı bilgi modelinde yer alan nesnelere ve nesneler arasındaki ilişkilere değinmektedir. Nesnelere ve nesne ilişkilerine yeni özellikler eklemek ve bunları manipüle edebilmek de hesaplamalı tasarım ile birlikte mümkün olmaktadır. Dynamo nesneler ile ilgili verileri YBM yazılımı olan Revit yazılımından alarak bunları işleyebilir ve YBM yazılımındaki nesneleri ve nesnelere ait verileri güncelleyebilen bir araç olarak kullanılabilir. YBM sisteminin geometrik temsiline yanında birçok geometrik olmayan veri barındırdığını düşünürsek hesaplamalı tasarımın mevcut YBM sistemi ile birlikte kullanılmasının verilere müdahale edilebilmesi ile birlikte getirdiği birçok avantaj meydana gelmektedir. Yapı bilgi modellerinde verinin proje süreci içerisinde artması ve karmaşıklaşması kullanıcıların bu süreci yönetmesinde yetersiz kalmasına ve hata sayısının artmasına sebep olmaktadır. Bu nedenle YBM tabanlı hesaplamalı tasarımın kullanımı oldukça ilgi görmektedir.

Ayrıca tesis yönetimi için YBM sistemlerinden faydalanılması ile birlikte bina tesis yönetimi için geometrik olmayan veriler giderek daha da önem kazanmaktadır. Bu aşama için YBM sistemlerinin faydaları çokça belgelenmiş olsa da YBM modelinin yeterliliği bu sistemin benimsenmesinde önemli engeller olarak karşımıza çıkmaktadır. Bu zorlukların üstesinden gelebilmek için YBM tabanlı hesaplamalı tasarım aracı Dynamo ile birlikte modelin bakım ve veri aktarım süreçleri otomatikleştirilmektedir. Dynamo'nun süreç içerisinde başarılı bir şekilde uygulanması, model bakımı için gerekli zamanı azaltarak tesis yöneticilerine gerekli bilgileri sağlamaktadır. Operasyonların ve bakımların toplam proje maliyetinin büyük bir kısmını oluşturduğu göz önüne alındığında, Dynamo'nun zaman ve maliyet üzerindeki potansiyel etkisi müşteriler ve proje sahipleri için cazip olmaktadır.

YBM tabanlı hesaplamalı tasarım aracı olan Dynamo genellikle YBM modeli içerisindeki geometrik ve geometrik olmayan veri manipülasyonu için kullanılmasının yanı sıra YBM modeli ile harici veri kaynakları arasındaki bilgi alışverişini sağlamak için de kullanılmaktadır. Dynamo'nun bu şekilde kullanılması bir sonraki bölüm olan bölüm 4.2.3'de detaylıca ele alınmaktadır.

Bu bölümde YBM tabanlı hesaplamalı tasarım aracı olan Dynamo'nun modelin yönetimi ve kalite kontrolü ile ilgili potansiyel uygulamaları ele alınmaktadır. Dynamo ile gerçekleştirilebilecek görevleri ana hatlarıyla; *projenin başlangıç ve ileri aşamalarında yer alan standardize işlerin gerçekleştirilmesi (1), modeldeki verilerin yönetilmesi ve doğruluğunun sağlanması (2) ve modeldeki verilerin görselleştirmesinin yapılması (3)* şeklinde üçe ayırmak mümkündür.

1. Revit ortamında yapılan standardize işlerin Dynamo ile birlikte çözülebileceği görevlerden bazıları aşağıda sıralanmıştır:

- Yapıya ait kat seviyelerinin (level) oluşturulması
- Yapıya ait kat planlarının oluşturulması
- Oluşturulan kat planlarının isimlendirilmesi
- Yapıya ait tavan kat planlarının oluşturulması
- Görünüşlerin oluşturulması

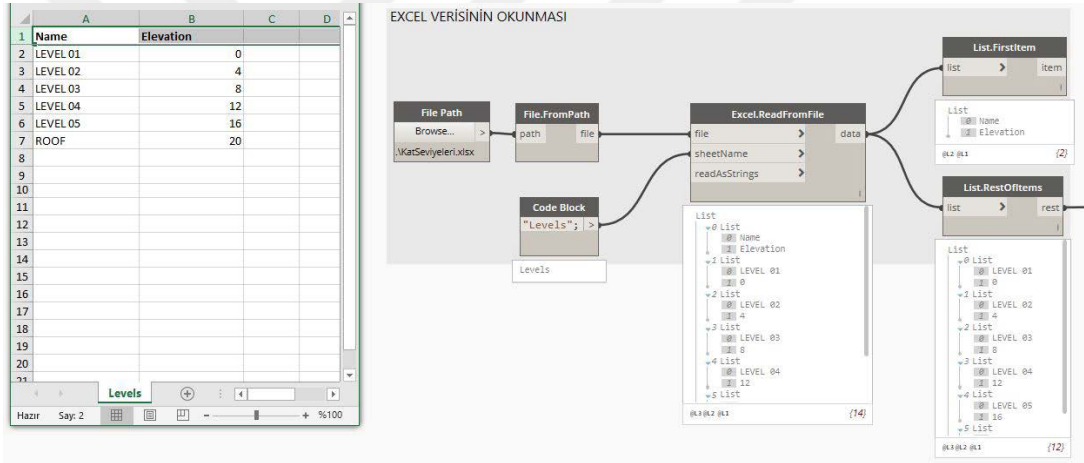
- Yapı kat planlarının ve görünüşlerinin paftalarının (sheet) oluşturulması
 - Oda numaralarının revize edilmesi
 - Belirlenen eğriye göre odaların numaralandırılması
 - Worksetlerin oluşturulması
 - Odalara göre duvar kaplama malzemelerinin oluşturulması
 - Odaların ve etiketlerinin, odaların merkezine getirilmesi
 - Autocad grid dosyasından Revit programında gridlerin oluşturulması
2. Yapı bilgi modelindeki verilerin yönetilmesi ve doğruluğunun sağlanması için Dynamo yazılımının gerçekleştirdiği görevlerden bazıları aşağıda sıralanmıştır:
- Tanımsız odaların modelden silinmesi
 - Projede yer alan grupların tekil nesnelere dönüştürülmesi
 - Revit programında yer alan nesnelerin ilgili worksetlere atanması
 - Nesnelerin hangi worksetlere ait olduğunun rapor edilmesi
3. Dynamo'nun modeldeki verilerin görselleştirmesini yaparak gerçekleştirdiği görevlerden bazıları aşağıda sıralanmıştır:
- Odaların görselleştirilmesi
 - Cephe tasarımı ve analizi
 - Mandrill paketi ile verilen görselleştirilmesi

4.2.2.1 Revit'te yapılması gereken standardize işlerin Dynamo ile yönetilmesi

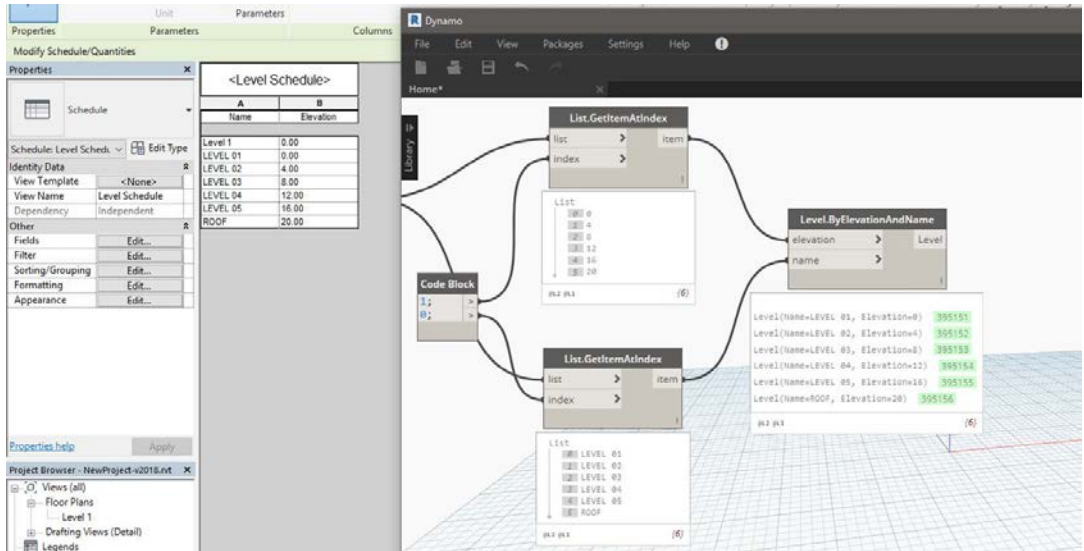
Bu bölümde, Revit ortamında yapılabilen fakat süreç içerisinde zaman ve emek kaybına sebep olan 12 tane iş akışının Dynamo ile nasıl daha verimli bir hale getirilebileceği uygulamalar üzerinden anlatılmaktadır.

Yapıya ait kat seviyelerinin (level) oluşturulması

YBM yazılımında manuel olarak yapılmakta olan kat seviyelerinin oluşturulma süreci için harcanan zaman ve emek, Dynamo yazılımında hazırlanan algoritma yardımıyla optimize edilebilmektedir. Özellikle yüksek katlı yapılar için projenin başlangıcında yapılması gereken bu işlem için harcanacak olan zaman kaybı önlenebilmektedir. Ayrıca bir kere hazırlanan kat seviyeleri oluşturma algoritması bütün projelere uygulanabilmektedir.



Şekil 4.61 : Kat yükseklikleri bilgisinin Excel'den alınması



Şekil 4.62 : Oluşturulan kat seviyelerinin (level) Revit'te görüntülenmesi

[illegible]

Yapıya ait kat planlarının oluşturulması

The screenshot displays a Power BI report with a table of floor plans and a complex DAX query diagram.

Table: Floor Plans

Name	Level
PLN_01	LEVEL 01
PLN_02	LEVEL 02
PLN_03	LEVEL 03
PLN_04	LEVEL 04
PLN_05	LEVEL 05
PLN_ROOF	ROOF
WORK_PLN_01	LEVEL 01
WORK_PLN_02	LEVEL 02
WORK_PLN_03	LEVEL 03
WORK_PLN_04	LEVEL 04
WORK_PLN_05	LEVEL 05
WORK_PLN_ROOF	ROOF

DAX Query Diagram:

The diagram illustrates the DAX query logic for the 'Floor Plans' table. It shows the following steps:

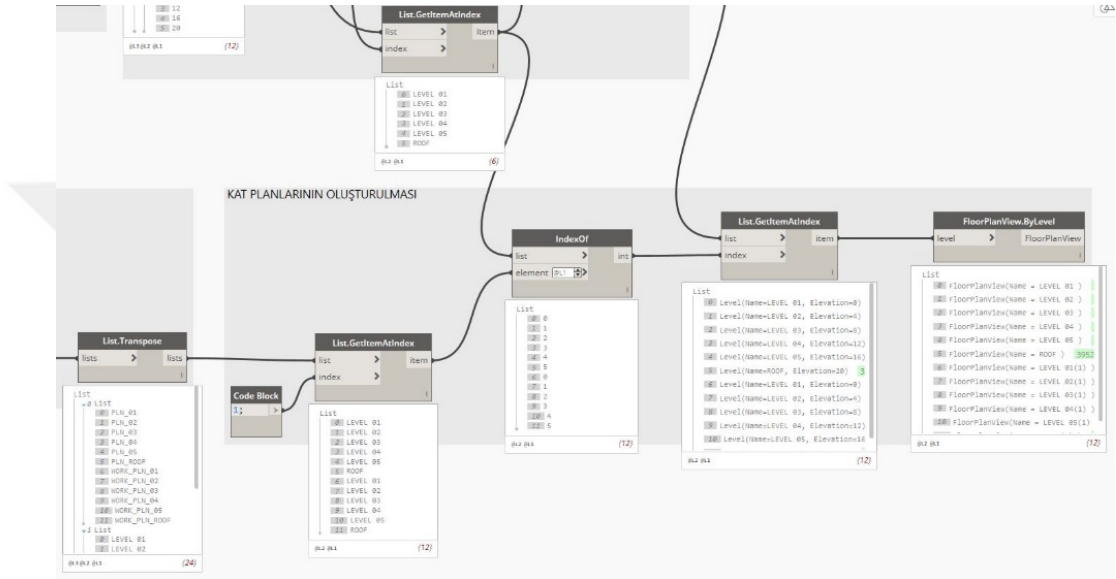
- File Path** (Browse...) and **Code Block** (Floor Plans) are used to define the **File From Path** (path) and **Excel Read From File** (file) parameters.
- The **Excel Read From File** (file) parameter is used to read the data from the Excel file, resulting in a **data** table.
- The **data** table is then processed by the **List First Item** (list) and **List Rest Of Items** (list) functions to extract the first item and the rest of the items, respectively.
- The **List Rest Of Items** (list) function is further processed by the **List Transpose** (list) function to transpose the data.

The final output of the query is a table with the following structure:

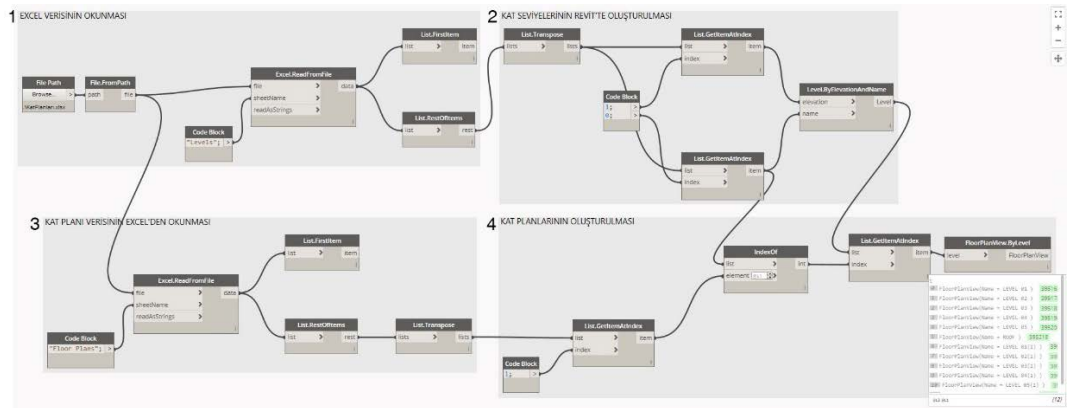
File Path	File From Path	Excel Read From File	List First Item	List Rest Of Items	List Transpose
File Path	File From Path	Excel Read From File	List First Item	List Rest Of Items	List Transpose

106

Kat planlarının oluşturulması için öncelikle Excel dosyasından hangi kat seviyesinde kaç adet kat planının yer alacağı bilgisi alınır (Şekil 4.64). Excel dosyasında yer alan kat planlarının isim bilgisi kat planları isimlendirilirken kullanılmaktadır. Kat planları için kat seviyesine dair bilgiler oluşturulan kat seviyesi listesinden alınır. “FloorPlanView.ByLevel” düğümü ile Excel’de tanımlanan 12 tane kat planı oluşturulmuş olur (Şekil 4.65) Oluşturulan kat planlarının ismi Level 01, Level 01(1) şeklindedir. Kat planlarının isimlendirilmesi için algoritma devam ettirilmiştir.



Şekil 4.65 : Kat planlarının oluşturulması



Şekil 4.66 : Kat seviyelerinin ve planlarının oluşturulması için yazılan algoritma

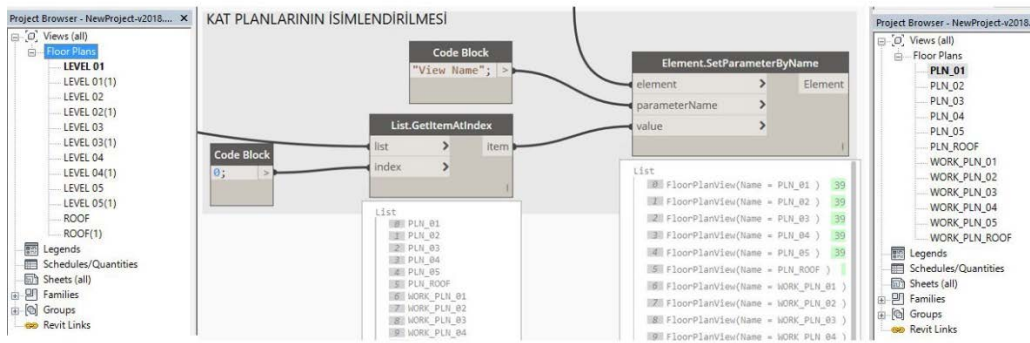
Dynamo’da oluşturulan algoritmanın tamamı Şekil 4.66’da yer almaktadır. Revit programında oluşturulan kat planları Dynamo düğümünün ön izlenmesinde yeşil renk ile ifade edilmektedir.

1. Kat seviyelerine ait verilerin Excel’den okunması

2. Kat seviyelerinin Revit'te oluşturulması
3. Kat planına ait verilerin Excel'den okunması
4. Kat planlarının Revit'te oluşturulması

Oluşturulan kat planlarının isimlendirilmesi

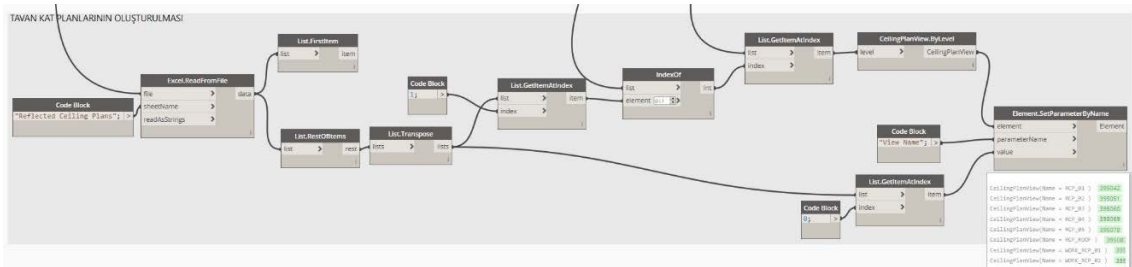
Revit'te oluşturulan kat planları, “Element.SetParameterByName” düğümü ile Excel tablosundaki kat planı verilerine göre yeniden isimlendirilmiştir (Şekil 4.67). Proje başlangıç aşamasında bu algorithmadan yararlanılacağı gibi projenin ilerleyen aşamalarında oluşturulan her kat planı ile birlikte artan veri yükünün beraberinde getirdiği harcanan zamanı azaltmak için de bu algoritma kullanılabilmektedir.



Şekil 4.67 : Kat planlarının yeniden isimlendirilmesi algoritması

Yapıya ait tavan kat planlarının oluşturulması

Yapıya ait kat planlarının oluşturulması için yazılan algoritmanın (Şekil 4.68) çalışma mantığı tavan kat planlarının oluşturulması için kullanılan algoritma ile aynıdır.

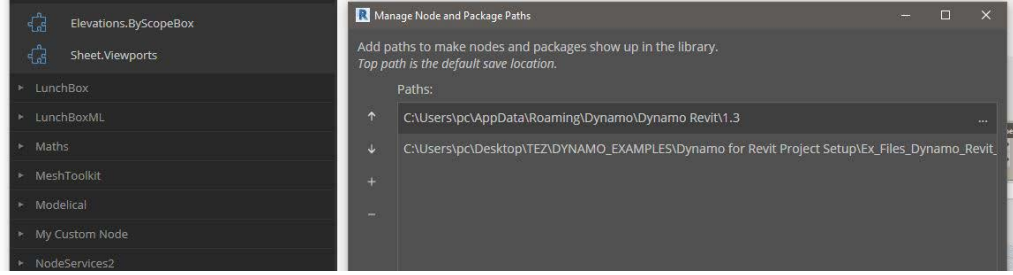


Şekil 4.68 : Tavan kat planlarının oluşturulması ve isimlendirilmesi için yazılan algoritma

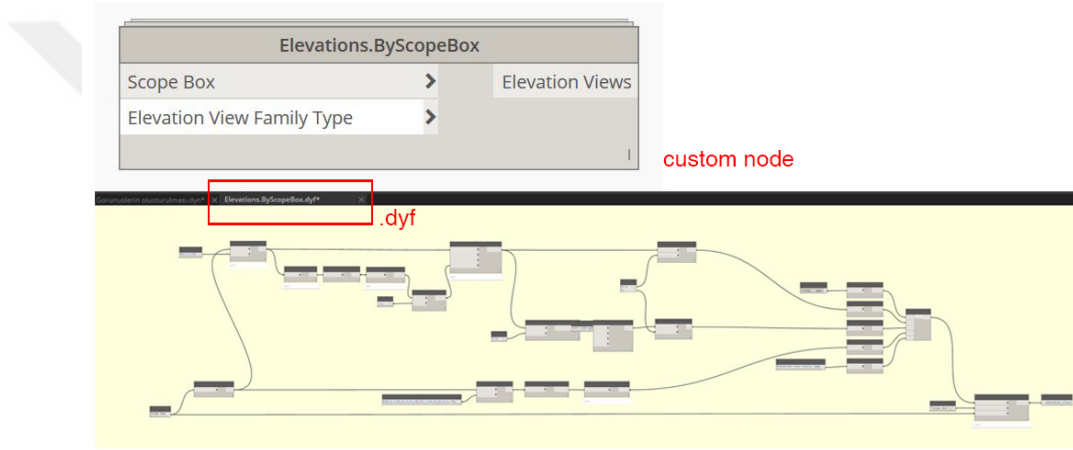
Görünüşlerin oluşturulması

Revit ortamında görüşlerin oluşturulması için hazırlanan algorithmada özel düğümlerden (custom node) yararlanılmıştır. Özel düğümler hazırlandıktan sonra Dynamo'nun özel düğümlerinin ve paketlerinin bulunduğu klasöre taşındığında

Dynamo kütüphanesi üzerinden hazırlanan özel düğümlere ulaşılabilmektedir (Şekil 4.69). Özel düğümlerin düzenlenmesi istenildiğinde .dyf uzantılı Dynamo algoritmasına ulaşılabilmektedir (Şekil 4.70)

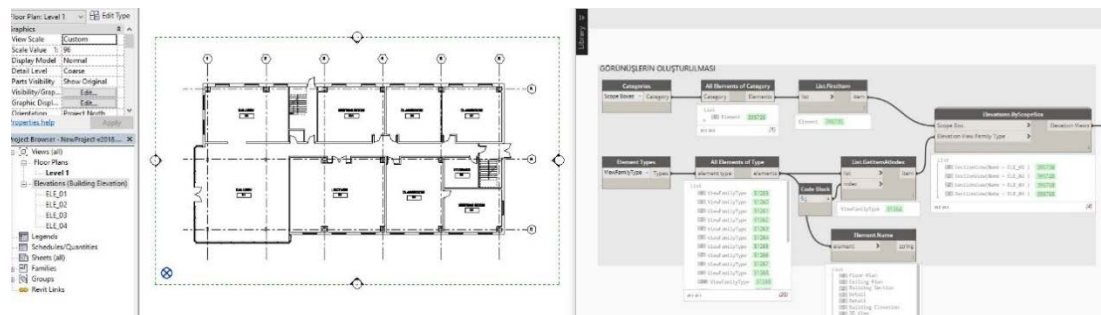


Şekil 4.69 : Özel düğümlerin Dynamo kütüphanesine yerleştirilmesi

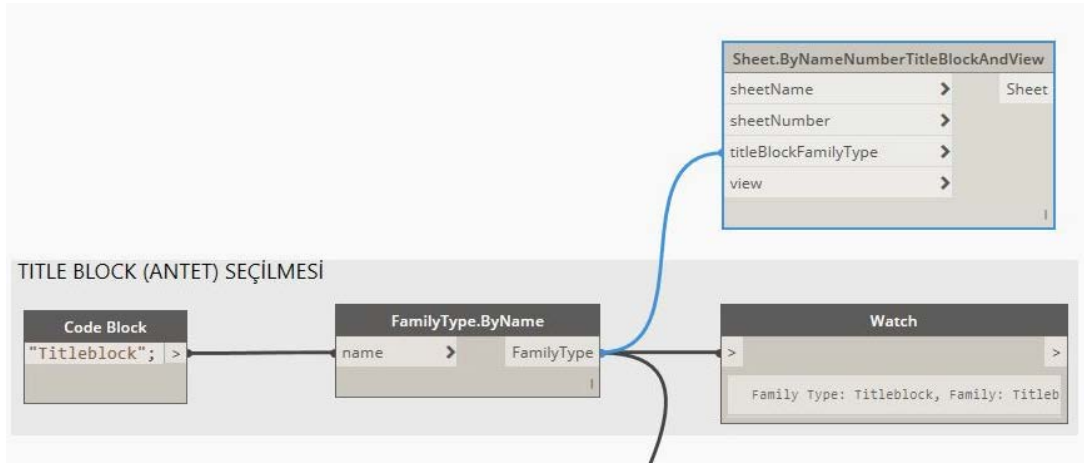


Şekil 4.70 : Özel düğümlerin düzenlenebilmesi

Görünümlerin oluşturulması görevi için “Elevations.ByScopeBox” özel düğümü kullanılmıştır. Bu düğüm için öncelikle Revit’te bir scope box (kapsam kutusu) çizilmesi gerekmektedir (Şekil 4.71). Algoritmanın yürütülmesi ile çizilen kapsam kutusunun her bir kenarına Revit’te yer alan görünüş sembolü (building elevation family) yerleştirilir. Böylece Revit’te dört adet görünüş elde edilmiş olur. Daha sonra bu görünüşler yeniden isimlendirilerek düzenlenir (Şekil 4.72).

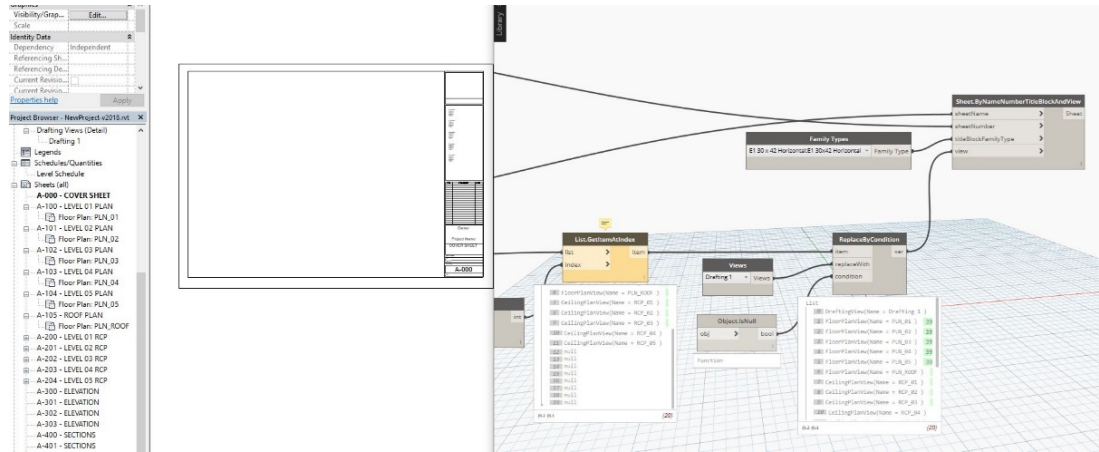


Şekil 4.71 : Revit’te scope box oluşturulması ve algoritmaya tanıtılması



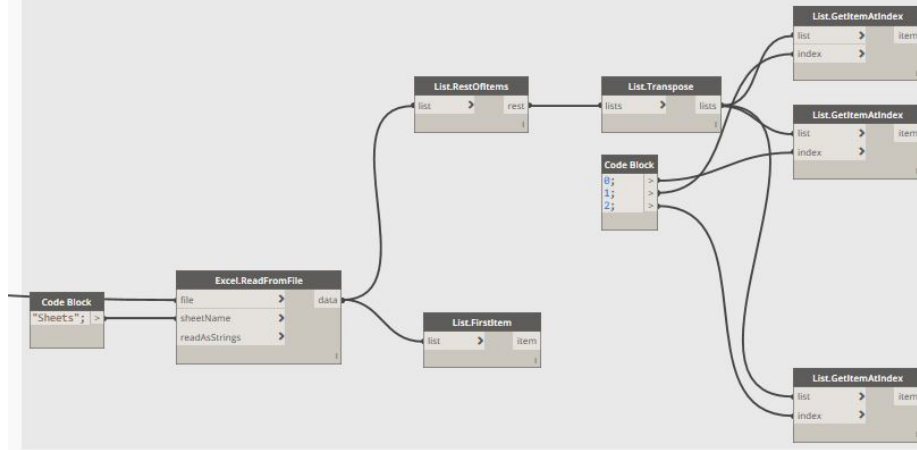
Şekil 4.74 : Revit ortamından antetin seçilmesi ve pafta oluşturmak için kullanılan düğüm

Dynamo programında hazırlanan algoritmanın yürütülmesi sonucu içerisinde plan bulunmayan başlangıç sayfası gibi paftalar oluşturulamamıştır. Çünkü Excel dosyasında plan sekmesi boş bırakılan değerler sıfır olarak algılanmış ve hazırlanan algoritmanın karşılığı olarak pafta üretilmemiştir. Bu durumda Dynamo ile hazırlanan algoritmanın çalıştırılması sonucu oluşan sıfır değerleri Revit dosyasında bulunan Drafting planı ile tanımlandığında Revit ortamında içerisinde planların bulunmadığı paftalar oluşturulabilmektedir (Şekil 4.75). Paftaların oluşturulması ve planların paftalara yerleştirilmesi ile ilgili hazırlanan algoritmanın tamamı Şekil 4.76’de yer almaktadır.

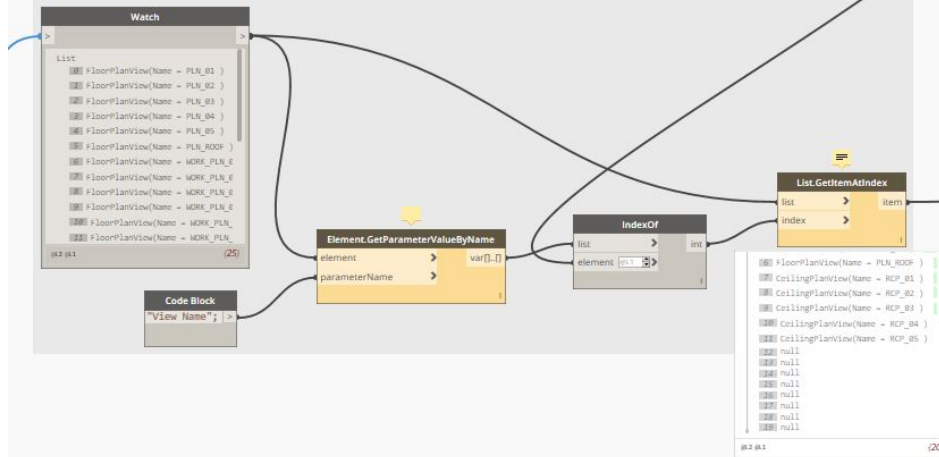


Şekil 4.75 : Revit ortamında bütün paftaların oluşturulması

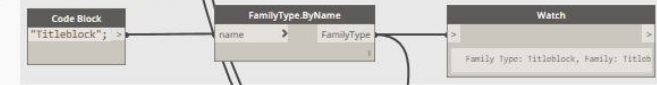
PAFTA BİLGİSİNİN EXCEL'DEN OKUNMASI



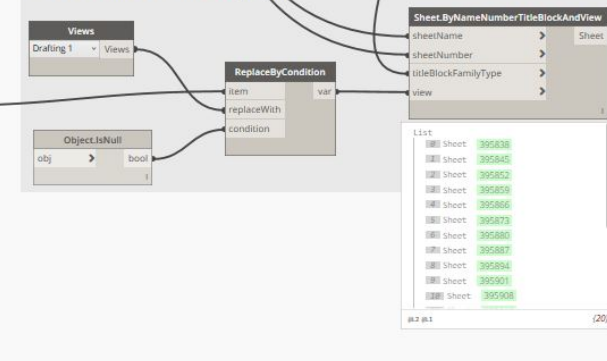
PLANLARIN VE GÖRÜNÜŞLERİN TOPLANMASI



TITLE BLOCK (ANTET) SEÇİLMESİ

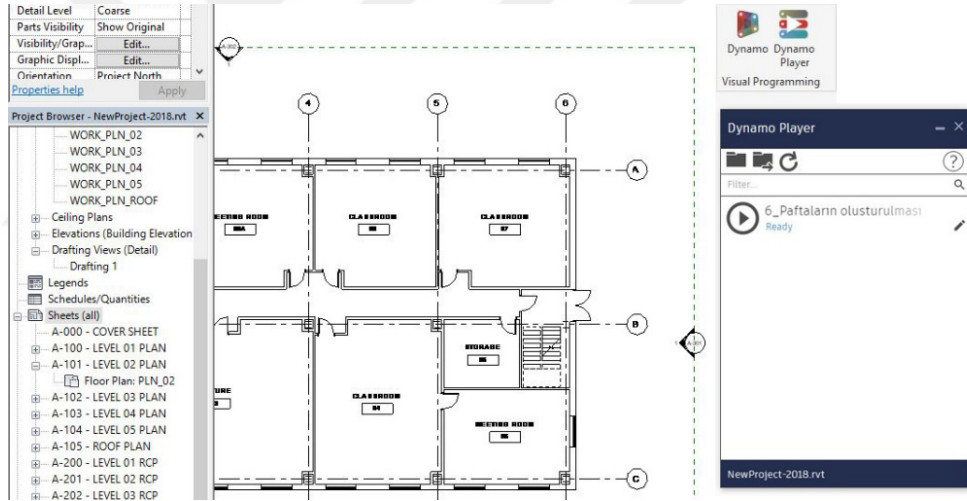


PLANLARI OLMAYAN PAFTALARI OLUŞTURMA

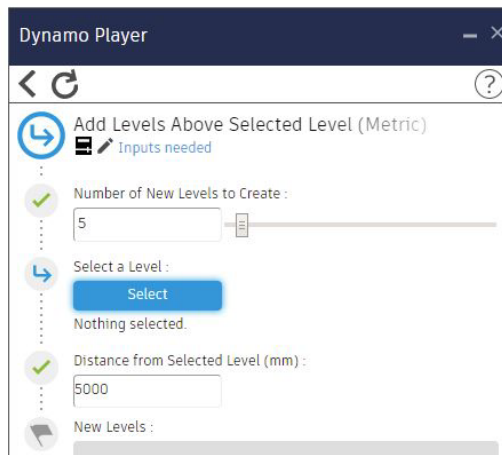


Şekil 4.76 : Pafta oluşturulması için hazırlanan algoritma

Dynamo ile hazırlanan algoritma kullanıcılar tarafından sadece DynamoPlayer üzerinden çalıştırılabilmektedir (Şekil 4.77). Dynamo Player, Revit programında Dynamo programını açmadan komut dosyalarını yürütmek için daha basit bir yol sağlamaktadır. Revit 2018.1 sürümü ile gelen yeni özellik sayesinde de Dynamo programını açmadan, Dynamo Player üzerinden algoritmaya ait girdi değerleri tanımlanabilmektedir (Şekil 4.78). Böylece, Dynamo kullanım bilgisi olmayan kullanıcılar için algoritmaların yürütülmesi oldukça kolay hale gelmektedir. Ayrıca Dynamo programının açılmadan çok daha kısa sürede algoritmanın yürütülebilmesi zamandan kazandırmaktadır. Tek bir klasörün içinde yer alan Dynamo algoritmaları, Dynamo Player ekranında alt alta sıralanabilmektedir. Böylece mimarlar ve mühendisler, ihtiyaç duyulan algoritmaları sınıflandırarak YBM tabanlı süreç içerisinde ofislerinde karşılaştıkları problemleri Dynamo Player ile birlikte kısa sürede optimize edebilmektedirler.



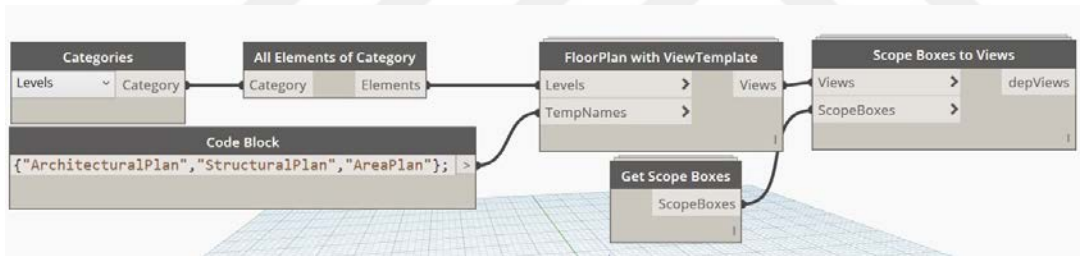
Şekil 4.77 : Dynamo Player ile hazırlanan algoritmanın yürütülmesi



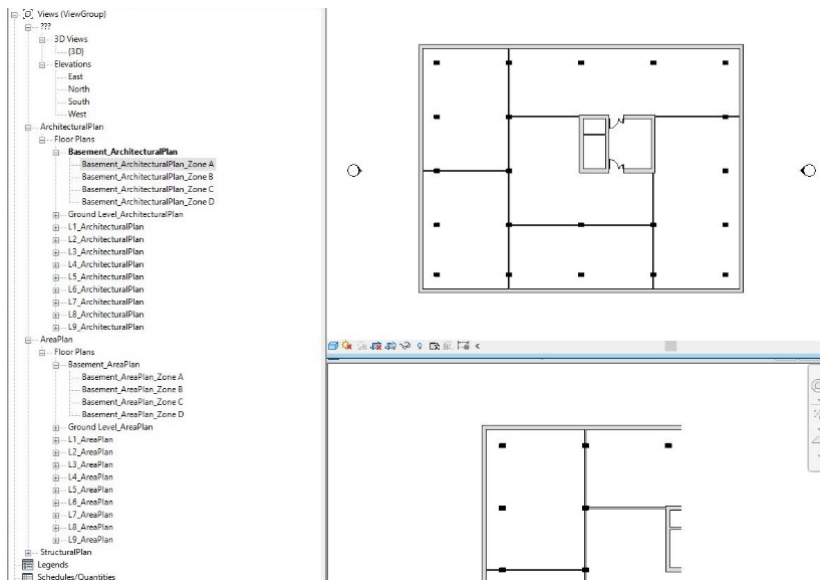
Şekil 4.78 : Dynamo Player ile girdilerin, Revit ortamında değiştirilebilmesi

Dynamo ortamında anlatıldığı şekilde kat planları oluşturulabileceği gibi bu görev için hazırlanmış özel düğümler de mevcuttur. Modelical paketine ait düğümlerle kat planları oluşturulabilmekte ve kat planlarına ilgili görünüm şablonu (view template) atanabilmektedir. Bunun için Modelical paketine ait “FloorPlan with ViewTemplate” düğümü ile “Scope Boxes to Views” düğümü kullanılmıştır (Şekil 4.79). Kat seviyesine göre planlar oluşturulurken Revit programında planlara görünüm şablonu atanmış ve planlar gruplandırılmıştır. Daha sonra kat planları büyük ölçekli planlara bölünmüştür (Şekil 4.80). Bunun için Revit’te oluşturulan alanı tanımlayan kutular (scope boxes) kullanılmıştır.

MMİ sektöründe yer alan mimarın tecrübesini, karar mekanizmasını gerektirmeyen bu tarz görevlerin Dynamo’da yapılması ile birlikte zaman kaybının önüne geçilmektedir. Örneğin, yapılan uygulamada projede yer alan 11 kat için mimari, statik, alan hesaplaması ile ilgili 3 farklı plan türü oluşturularak her bir kat planı kendi içinde 4 ayrı bölgeye bölünmüştür. Oluşturulması gereken 132 kat planı Dynamo’da hazırlanan algoritma ile birlikte birkaç saniye içerisinde oluşturulmuştur.



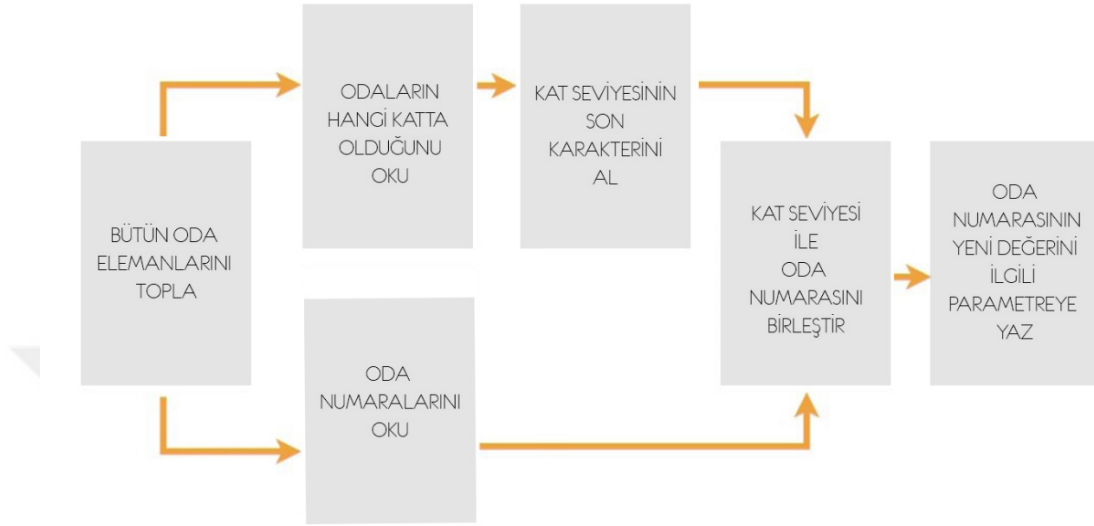
Şekil 4.79 : Dynamo’da hazırlanan kat planlarının oluşturulması algoritması



Şekil 4.80 : Dynamo’da hazırlanan kat planlarının Revit’te görüntülenmesi

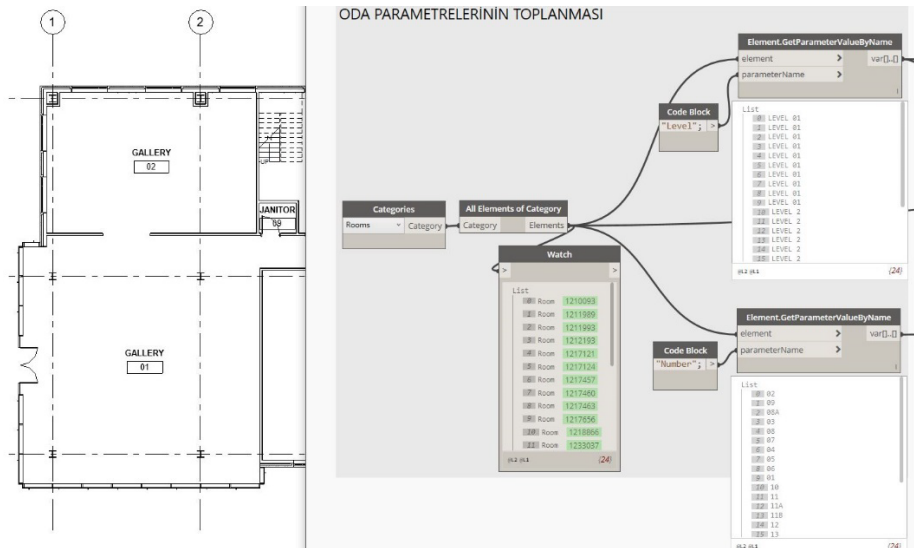
Oda numaralarının revize edilmesi

Revit programı ile hazırlanan projede yer alan oda numaralarının revize edilmesi konusu bir uygulama ile ele alınmaktadır. Dynamo ortamında hazırlanan algoritmanın mantığı Şekil 4.81’de gösterilmektedir.

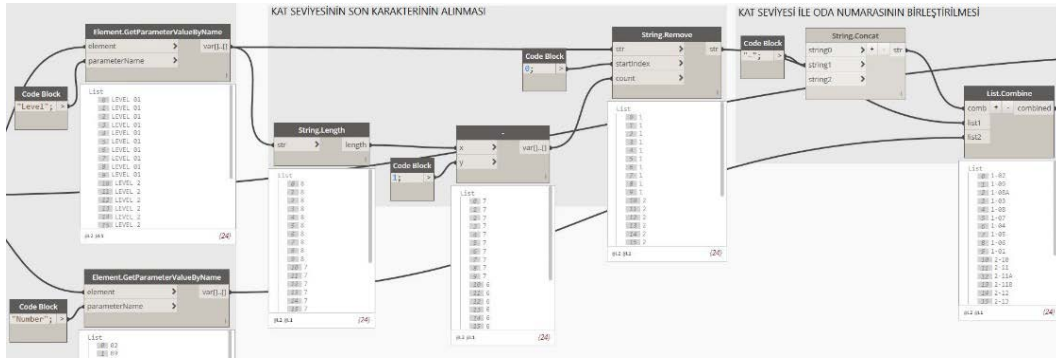


Şekil 4.81 : Oda numaralarının nasıl revize edileceğine dair hazırlanan algoritmanın çalışma mantığı

Öncelikle Revit ortamında oluşturulmuş olan yapı bilgi modelinde yer alan odaların isimlendirilmesine ait numaralar ve bulunduğu kat bilgileri alınır (Şekil 4.82). Revit’te yer alan odalardan bu veriler alındıktan sonra odaların hangi katta bulunduğunu temsil eden rakamlar bu verilerden ayıklanır. Daha sonra bu verilerle odaların isimlerini temsil eden numaralar birleştirilir ve odaların yeni numaraları elde edilmiş olur (Şekil 4.83).

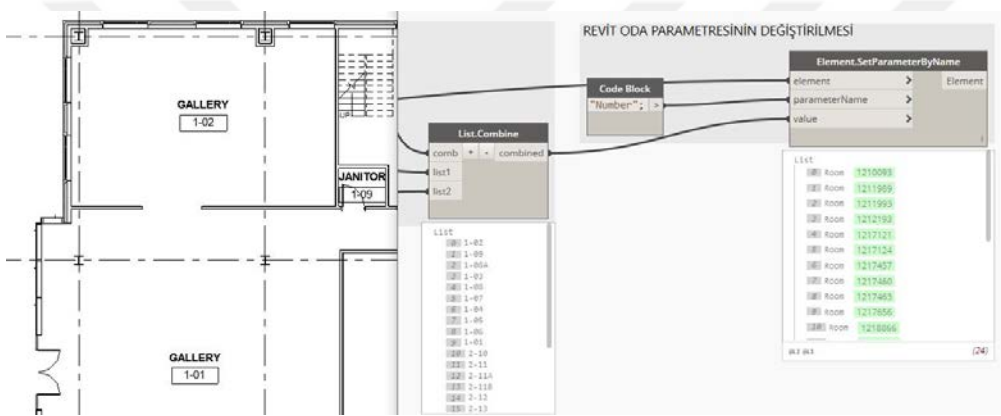


Şekil 4.82 : Oda parametrelerinin Revit modelinden alınması



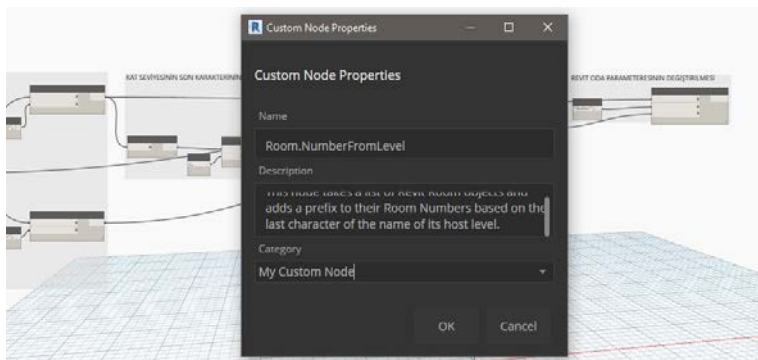
Şekil 4.83 : Odaların yeni numaralarının oluşturulması

Dynamo ortamında elde edilen bu parametreler Revit ortamına aktarılır. Odanın özellikleri menüsünde yer alan ilgili parametre Dynamo tarafından otomatik olarak değiştirilir (Şekil 4.84).

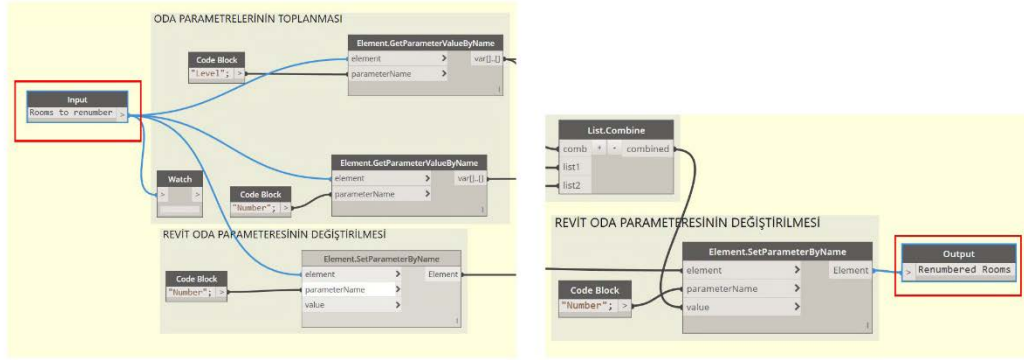


Şekil 4.84 : Odaların numaralarının Revit'e aktarılması

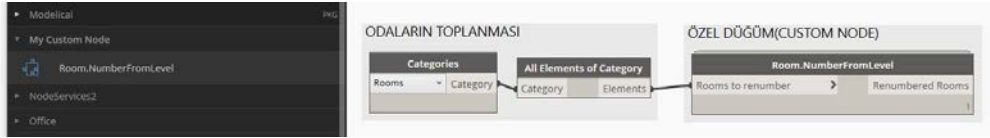
Dynamo ortamında hazırlanan bu balgoritma daha sonraki projelerde de kullanılmak üzere özel düğüm (custom node) olarak kaydedilebilir (Şekil 4.85). Hazırlanan algoritma özel düğüm olarak kaydedilirken gerekli girdi ve çıktıların tanıtılması gerekmektedir. Bu örnek için odaların hangi katta bulunduğu ve mevcut oda numaraları girdi olarak tanımlanırken revize edilmiş oda numaraları çıktı olarak tanımlanmıştır (Şekil 4.86).



Şekil 4.85 : Özel düğüm oluşturma ekranı



Şekil 4.86 : Özel düğüm için girdi ve çıktılar tanımlanması

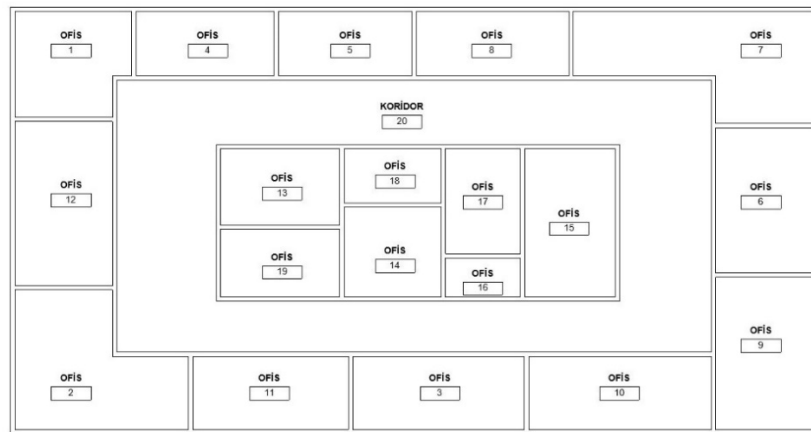


Şekil 4.87 : Oluşturulan özel düğüm ile algoritmanın hazırlanması

Oluşturulan özel düğüm, Dynamo kütüphanesine kaydedilebilmektedir (Şekil 4.87). Böylece kullanıcılar rahatlıkla kendi hazırladıkları düğümü kullanabilmekte ve diğer kullanıcılarla paylaşabilmektedirler.

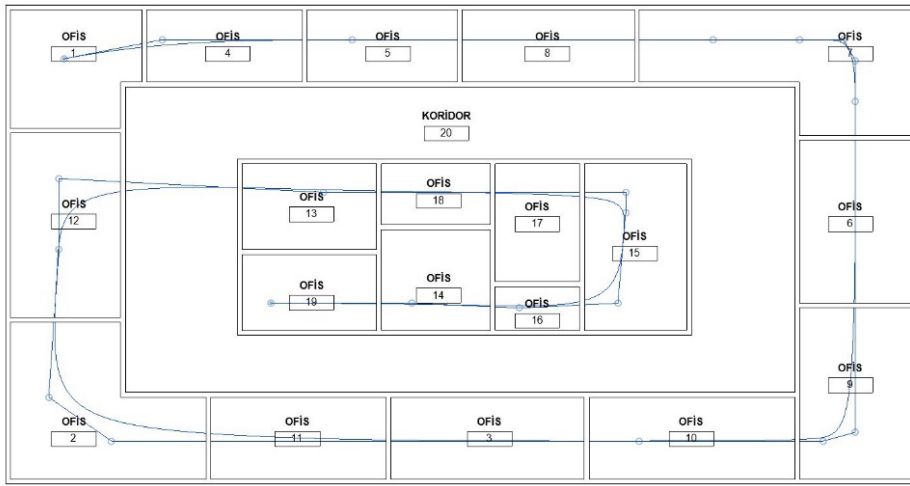
Belirlenen eğriye göre odaların numaralandırılması

Proje ilerlerken kat planlarında yer alan odalar güncellendikçe oda numaralarının yapılan değişikliklere ve proje kriterlerine bağlı kalınarak değiştirilmesi gerekmektedir. Her değişiklikte oda numaralarının manuel olarak değiştirilmesi, oldukça zaman alan bir iş yükü olmaktadır. Bu tür bir düzenleme, Excel’de bir tablo biçiminden veya bir Revit çizelgesi üzerinden kolayca manipüle edilememektedir. Oda numaralarını yeniden numaralandırılması uygulamasının Dynamo ile nasıl yapılacağı bir kat planındaki odaların mekânsal düzeni üzerinden anlatılmaktadır (Şekil 4.88).

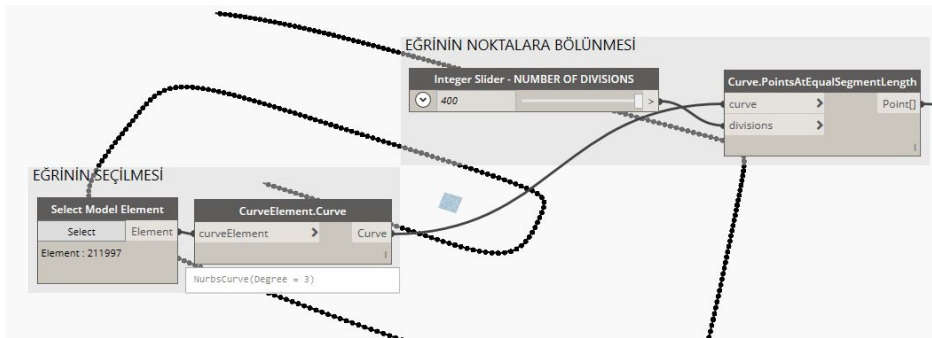


Şekil 4.88 : Revit’te oda numaraları sıralı olmayan kat planı

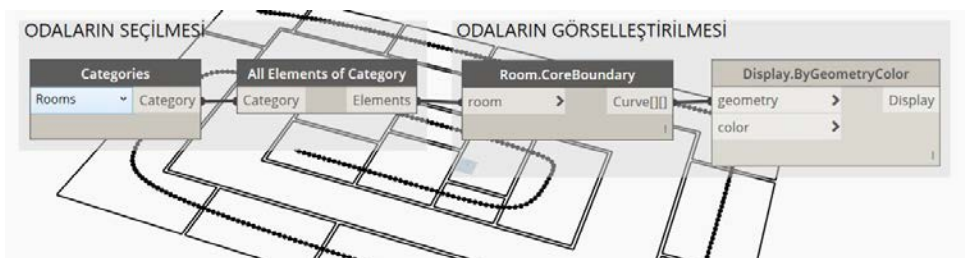
Bu sorunu çözmek için, kullanıcıların Revit programında numaralandırılacak odalardan geçen bir eğri çizmeleri gerekmektedir (Şekil 4.89). Sadece eğri ile kesişen odalar yeniden numaralandırılmaktadır. Revit programında çizilen eğri Revit seçim araçları ile birlikte Dynamo içerisine alınır ve belirli noktalara bölünür (Şekil 4.90). Daha sonra Revit programında yer alan odalar Dynamo'ya tanıtılır ve istenildiğinde görselleştirilebilir (Şekil 4.91). Bir sonraki adım olarak Dynamo'ya tanıtılan odalar ile eğrilerin kesişimi alınır ve eğri ile kesişen odaların listesi çıkartılır (Şekil 4.92). Listesi çıkartılan odalar belirtilen kriterlere ve çizilen eğriye göre yeniden numaralandırılır (Şekil 4.93).



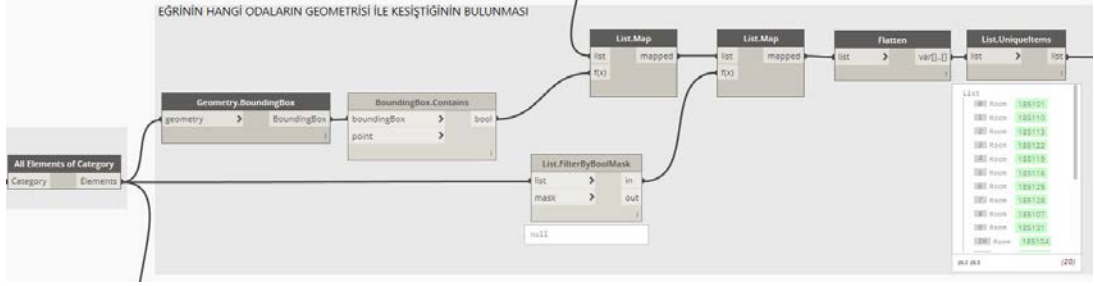
Şekil 4.89 : Revit programında eğrinin çizilmesi



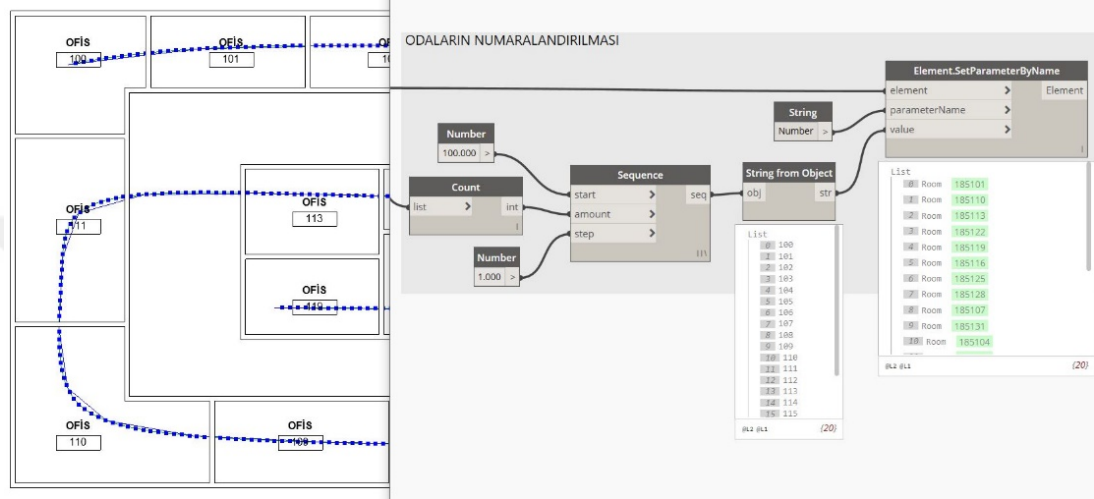
Şekil 4.90 : Dynamo ile eğrinin seçilmesi



Şekil 4.91 : Dynamo ile odaların seçilmesi ve görselleştirilmesi

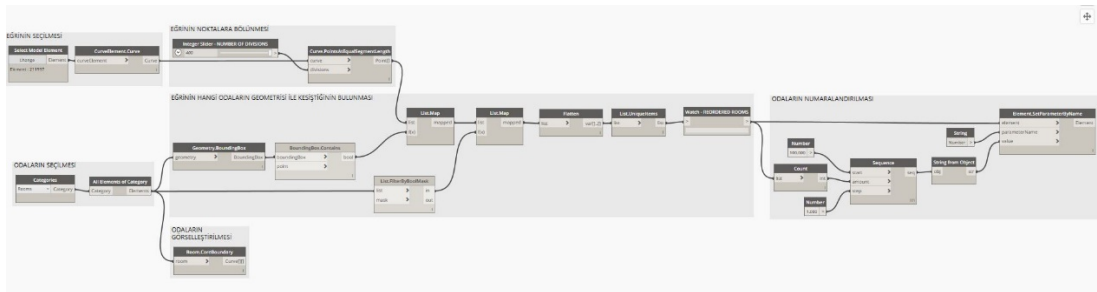


Şekil 4.92 : Dynamo’da eğri ve odaların kesiştirilmesi



Şekil 4.93 : Odaların çizilen eğriye göre yeniden numaralandırılması

Dynamo programında hazırlanan bu algoritmanın yürütülmesi, Dynamo Player ile otomatikleştirilebilmektedir. Proje revize edildiğinde oda numaralarının yeniden düzenlenmesi için harcanan zaman yazılan algoritma ile minimize edilmektedir. Algoritmanın tamamı Şekil 4.94’de yer almaktadır.



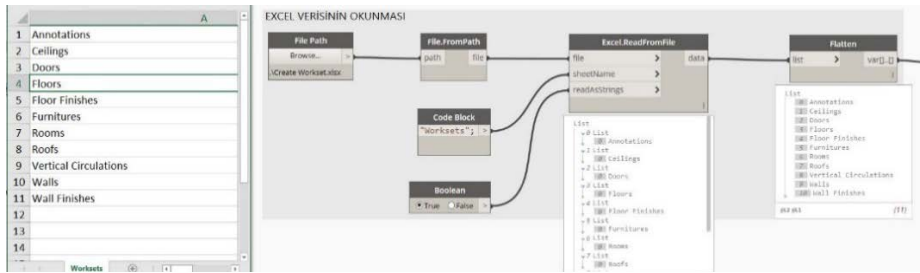
Şekil 4.94 : Odaların yeniden numaralandırılması için hazırlanan algoritma

Bir eğri ile numaralandırma işleminin yapılması başka problemlerin çözümü için de kullanılabilir. Otoparktaki park alanlarının isimlendirilmesi için de bu algoritmanın kullanılması oldukça yararlı olmaktadır.

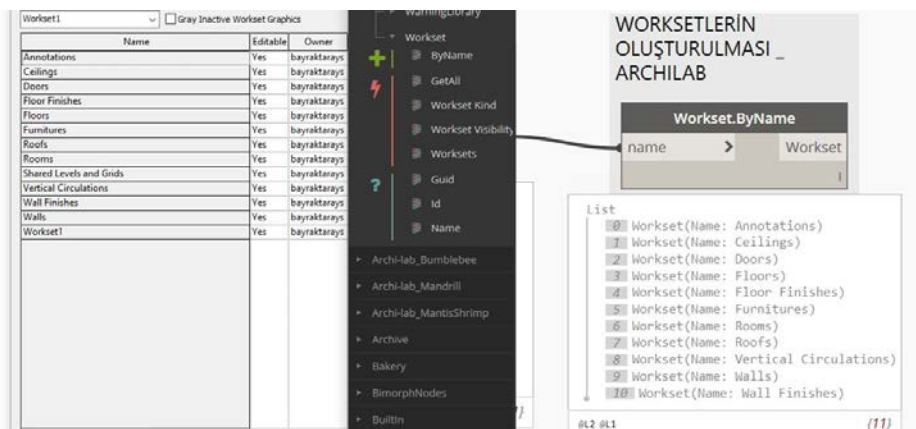
Revit programında worksetlerin oluşturulması

Revit başlangıç ayarı olarak projeye ait worksetlerin oluşturulması gerekmektedir. Bu işlemin her proje için ayrı ayrı yapılması gerekmektedir. Projenin karmaşıklığına ve büyüklüğüne bağlı olarak projede yer alan workset sayısı artabilmektedir. Ayrıca ana modele linklenen her dosya için de ayrı bir workset oluşturulması gerekmektedir. Bu nedenle workset oluşturulması işi zaman alan bir iş kalemi olabilmektedir. Bu nedenle Dynamo ile bu iş için harcanan zaman ve emeğin minimize edilmesi gerekli görülmektedir.

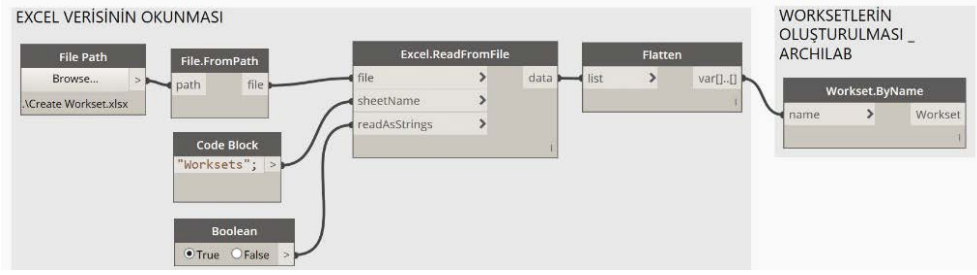
Revit programında workset oluşturmak için öncelikle, Excel’de bir workset listesi hazırlanır ve Dynamo’ya tanıtılır (Şekil 4.95). Elde edilen workset listesi, Archilab paketine ait “Workset.ByName” düğümüne girdi olarak tanıtılır. Böylece istenilen worksetler, Revit programında elde edilmiş olur (Şekil 4.96). Algoritmanın tamamı Şekil 4.97’de yer almaktadır. Dynamo programının açık kaynak olması sebebiyle birçok geliştirici özel problemlere yönelik çözüm üretmek amacıyla paketler ve düğümler geliştirmektedir. Archilab paketine ait “Workset.ByName” düğümü bunlardan sadece bir tanesidir. Dynamo’nun Revit ile karşılaşılan problemlere çözüm getirilmesi yönünde kendini geliştirmesi nedeniyle MMİ sektörü içerisinde kullanımı gittikçe artmaktadır.



Şekil 4.95 : Dynamo ile Excel’de hazırlanan workset verisinin okunması



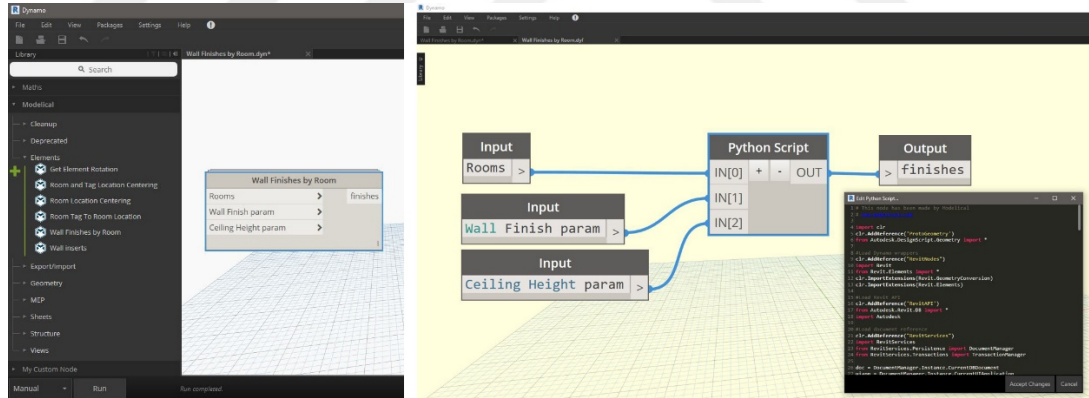
Şekil 4.96 : “Workset.ByName” düğümü ile worksetlerin oluşturulması



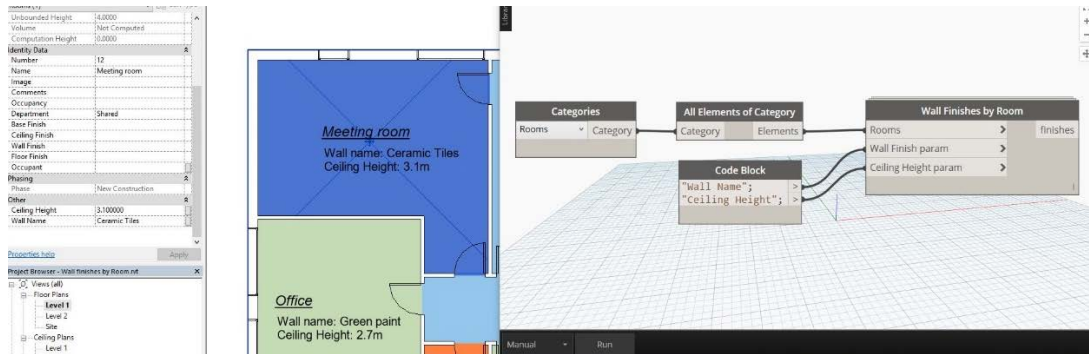
Şekil 4.97 : Worksetlerin oluşturulması için hazırlanan algoritma

Odalara göre duvar kaplama malzemelerinin oluşturulması

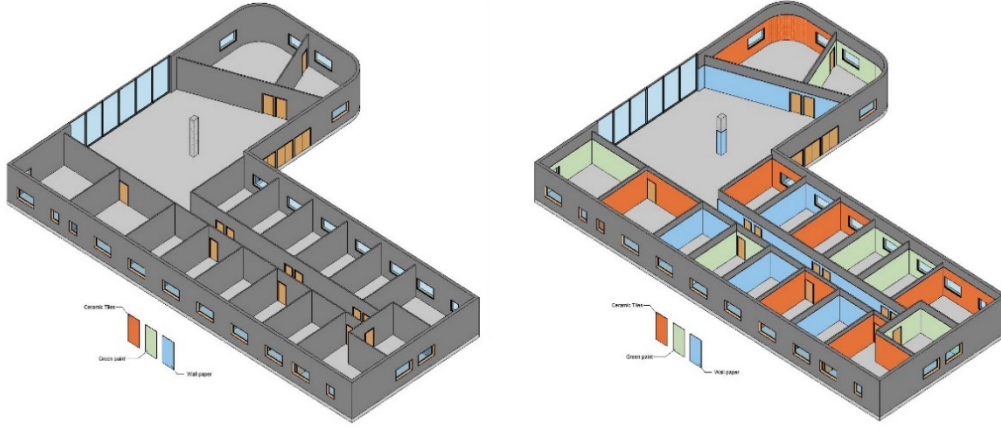
Bu örnekte, Revit yapı modelinde yer alan odalara ait parametrelere göre duvar kaplamasının Dynamo ortamında nasıl oluşturulabileceği ele alınmaktadır. Dynamo paketlerinden Modelical adlı paket kullanılmıştır. Modelical paketinde yer alan oda parametrelerine göre duvar kaplaması oluşturmak için geliştirilen “Wall Finishes by Room” adlı düğüm kullanılmıştır (Şekil 4.98). Dynamo’da oluşturulan algoritma için öncelikle bütün odalar Revit ortamından alınmıştır (Şekil 4.99). Daha sonra odalara ait olan duvar kaplama isimleri ve tavan yüksekliği bilgisi Dynamo ortamına aktarılmış ve Modelical düğümü ile odalar için belirtilen parametrelere göre duvar kaplamaları oluşturulmuştur (Şekil 4.100).



Şekil 4.98 : Modelical Paketi, “Wall Finishes by Room” düğümü ve açılımı



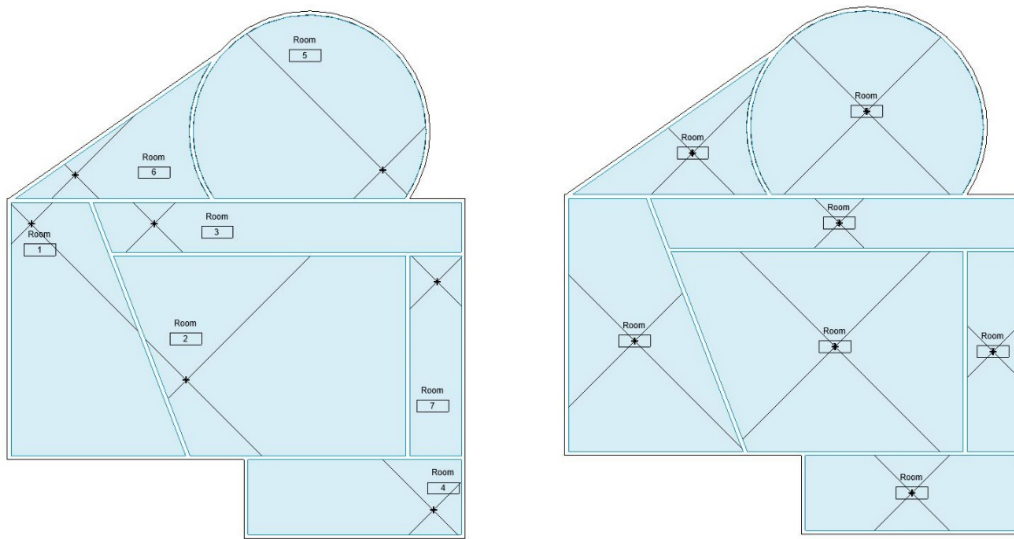
Şekil 4.99 : Revit programında oluşturulan odalar ve Dynamo programında hazırlanan algoritma



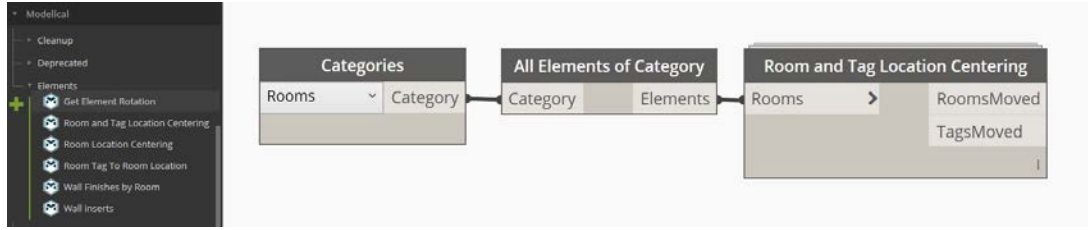
Şekil 4.100 : Duvar kaplamalarının oluşturulması

Odaların ve etiketlerinin, odaların merkezine getirilmesi

Bu örnekte, Revit'te yer alan odaların ve etiketlerinin, odaların merkezine getirilebilmesi için Dynamo'da algoritma hazırlanmıştır. Projenin büyüklüğüne bağlı olarak belgeleme ve okunabilirliğin artırılması amacıyla odaların etiketlerinin belirli bir düzende olması gerekmektedir. Revit programında etiketlerin odanın merkezine getirilmesi işi manuel olarak yapılmaktadır. Bu durum da zaman ve emek kaybına yol açmaktadır. Bu probleme çözüm olarak Modelical paketi özel bir düğüm geliştirmiştir. Oluşturulan “Room and Tag Location Centering” düğümü ile sadece Revit'te yer alan odalar seçilmekte ve algoritma yürütülmektedir (Şekil 4.102). Algoritmanın yürütülmesi ile odalar ve etiketleri, odaların merkezine getirilmektedir (Şekil 4.101).



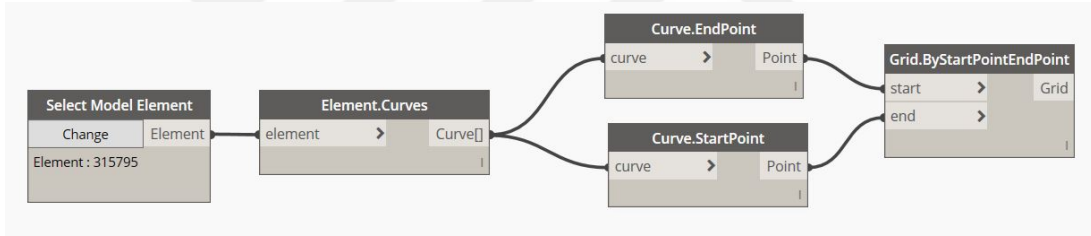
Şekil 4.101 : Odaların ve etiketlerinin merkeze getirilmesi işlemi



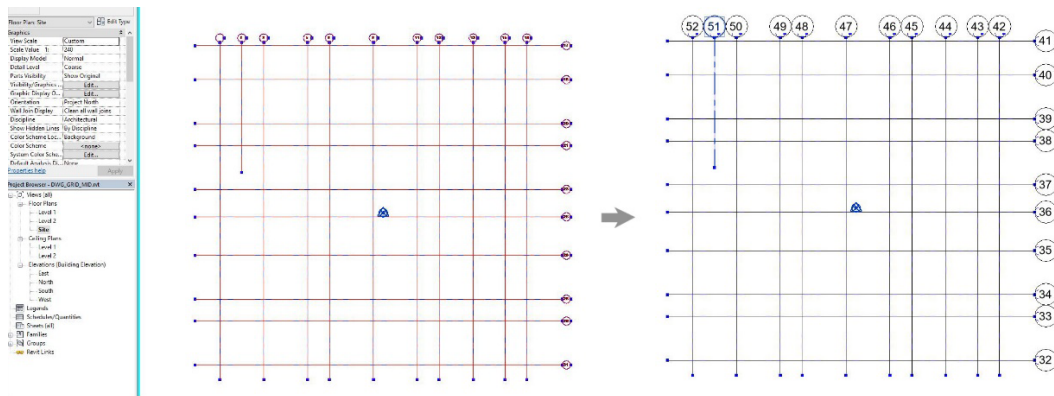
Şekil 4.102 : Dynamo’da hazırlanan algoritma

Autocad grid dosyasından Revit programında gridlerin oluşturulması

Birçok farklı disiplin ile birlikte çalışırken proje sürecinde farklı formatta dosya paylaşımı söz konusu olabilmektedir. Örneğin projeye başlarken gridler *dwg* formatında olabilir. *Dwg* formatında olan gridler, Revit ortamına aktarılırken Dynamo’dan yararlanılabilmektedir. Bunun için öncelikle *dwg* dosyası Revit programına alınır. Daha sonra Dynamo ile *dwg* dosyası seçilir ve hazırlanan algoritma yürütülür (Şekil 4.103). Algoritmanın yürütülmesi sonucunda gridler, Revit programında oluşturulmuş olmaktadır (Şekil 4.104).



Şekil 4.103 : Dynamo’da hazırlanan grid oluşturulması algoritması



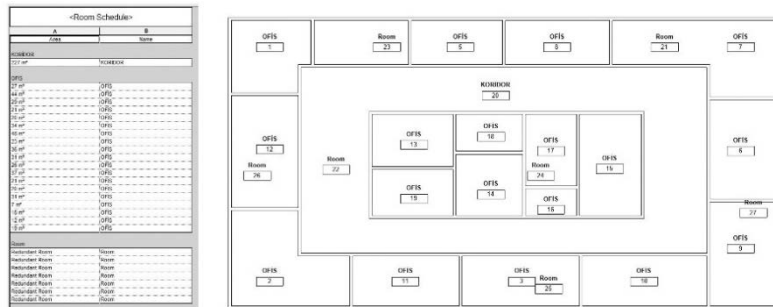
Şekil 4.104 : Dynamo ile birlikte gridlerin oluşturulması

4.2.2.2 Yapı bilgi modelindeki verilerin, Dynamo ile yönetilmesi ve doğruluğunun sağlanması

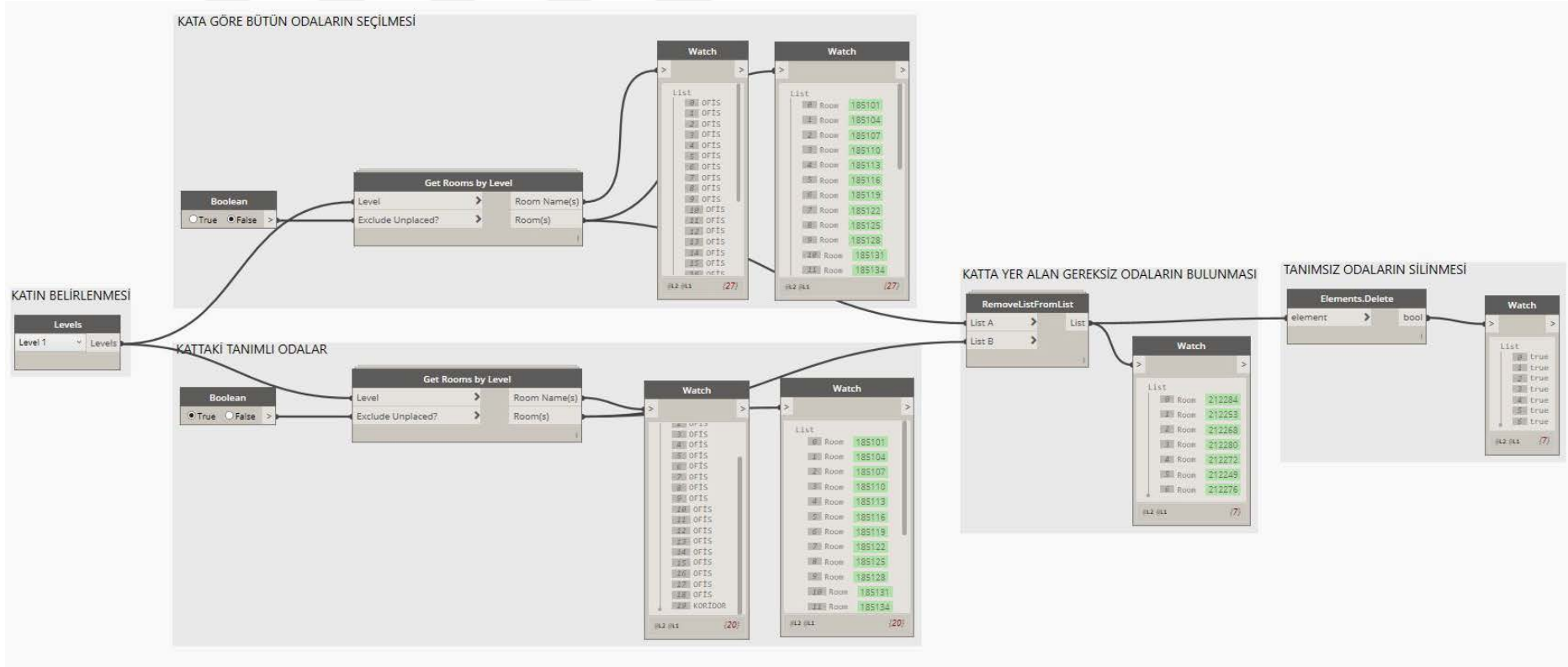
Geleneksel 2D çizim teknikleri ile kıyaslandığında mevcut YBM sistemleri proje süreci içerisinde meydana gelen değişikliklerden kaynaklanan iş yükünü optimize etmektedir; ancak, proje büyüklüğüne bağlı olarak yapılan değişiklikleri modele aktarmak hala çok fazla zaman ve emek gerektirmektedir. Harcanan zamanın ve emeğin minimize edilmesi için hesaplamalı tasarım aracı olan Dynamo'dan yararlanılmaktadır. Dynamo ile yapı bilgi modelinde yer alan gereksiz veriler, nesneler ve bileşenler temizlenebilmektedir. Bu işlem, farklı disiplinler ile düzenli olarak yapılan model teslimleri için önemli olmaktadır. Dynamo ile modeldeki nesneler ve bileşenler ait olmaları gereken worksetlere atanabilmektedir. Revit ile workset ataması yapılırken nesne gruplarının ayrı ayrı seçilmesi gerekmektedir. Dynamo ile tanımlanan bu problemin çözümü için harcanan zaman minimize edilmektedir. Ayrıca nesnelere ait isim, malzeme bilgisi, üretici bilgisi gibi parametrelerin değişmesiyle birlikte modelin güncellenmesi gerekmektedir. Dynamo ile bu güncelleme yapılabilmektedir. 4D simülasyon için gerekli olan nesnelere parametre ve lokasyon bilgisi tanımlama görevi de Dynamo ile birlikte başarı ile yürütülebilmektedir.

Tanımsız odaların silinmesi

Revit programında oda revizeleri sırasında bazen birden çok oda üst üste çıkışabilmektedir. Bu durum Revit'te gereksiz odaların olmasına sebep olmaktadır (Şekil 4.105). Gereksiz odaların silinmesi için Dynamo programında algoritma yazılmıştır. Bu algoritma için Archi-lab paketindeki düğümlerden yararlanılmıştır. Önce istenilen kattaki bütün odalar toplanır. Daha sonra tanımlı odaların listesi çıkartılır. Bütün odaların listesinden, tanımlı odaların listesi çıkartıldıktan sonra tanımsız odalar elde edilir ve elde edilen tanımsız odalar Dynamo ile Revit programından silinir (Şekil 4.106).



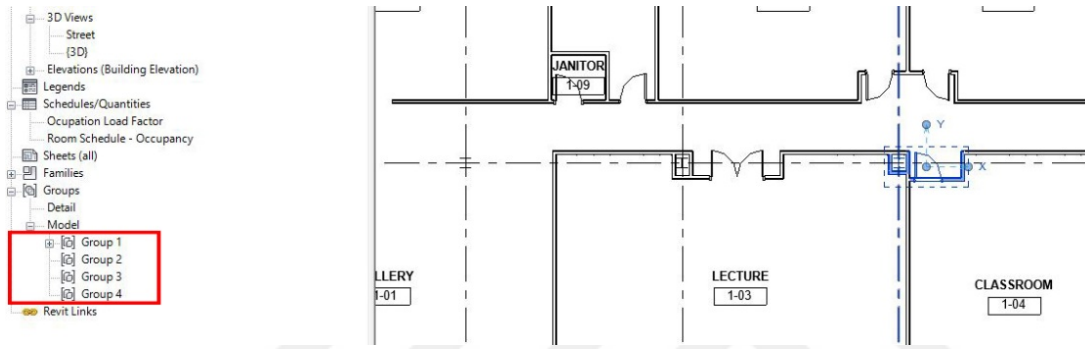
Şekil 4.105 : Revit programında yer alan gereksiz odalar



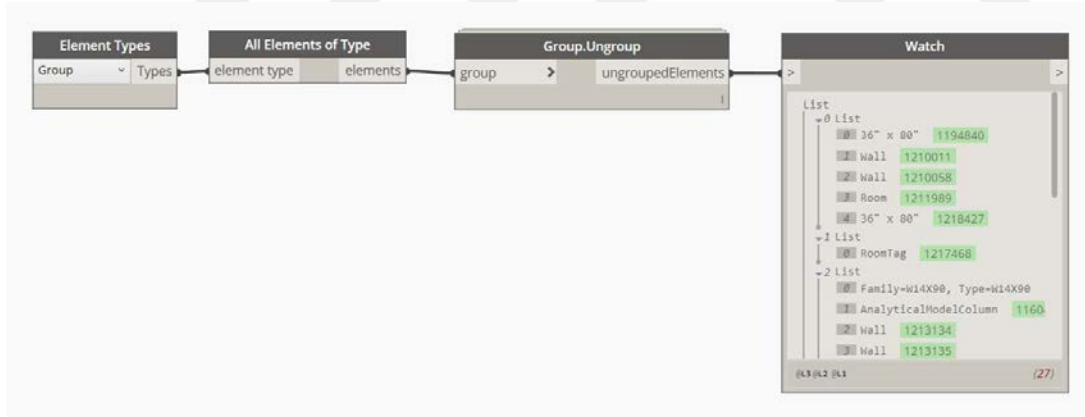
Şekil 4.106 : Dynamo’da hazırlanan gereksiz odaların silinmesi algoritması

Projede yer alan grupların tekil nesnelere dönüştürülmesi

Projenin farklı disiplinlerle paylaşımı ve model teslimi zamanında modelde yer alan nesnelerin grup halinde olması istenmemektedir. Her nesnenin ilgili worksetinde ve tekil bir biçimde bulunması çalışma açısından daha uygun olmaktadır. Bu nedenle Revit programında çalışma sırasında oluşturulan grupların tekil nesnelere dönüştürülmesi için Dynamo’da algoritma hazırlanmıştır. Dynamo ile Revit programında yer alan bütün gruplar toplanır (Şekil 4.107). Daha sonra Clockwork paketinde yer alan “Group.Ungroup” özel düğümü kullanılarak bütün gruplar kendi içerisinde yer alan tekil nesnelere dönüştürülür (Şekil 4.108).



Şekil 4.107 : Revit programında yer alan gruplar

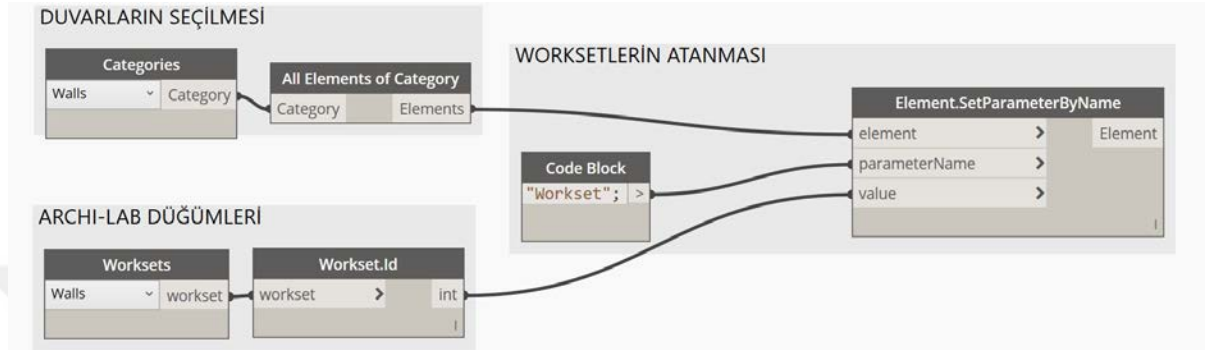


Şekil 4.108 : Dynamo’da grupların tekil nesnelere dönüştürülmesi için hazırlanan algoritma

Revit programında yer alan nesnelerin ilgili worksetlere atanması

Revit nesnelerinin doğru worksetlerde olması ile birlikte ekibin iş akışı daha iyi yönetilebilmektedir. Aynı zamanda farklı disiplinler ile birlikte çalışırken olması gereken işbirliğinin verimliliğini artırırken, kullanıcının model üzerindeki hakimiyetini de artırmaktadır. Revit programı ile nesneler, worksetlerine manuel olarak atanırken Dynamo ile birlikte nesneler kategorilerine göre gruplandırılarak

workset ataması yapılabilmektedir. Dynamo ile birlikte öncelikle duvar kategorisine ait bütün nesneler toplanır. Daha sonra Archi-lab paketine ait “Workset ve Workset.Id” düğümleri ile ilgili workset seçilir. Algoritmanın son adımı olarak seçilen nesneler ilgili worksete “Element.SetParameterByName” düğümü ile atanır (Şekil 4.109). Dynamo’da hazırlanan bu algoritma döşeme, pencere, kapı gibi diğer kategoriler için de tekrarlanabilmektedir.



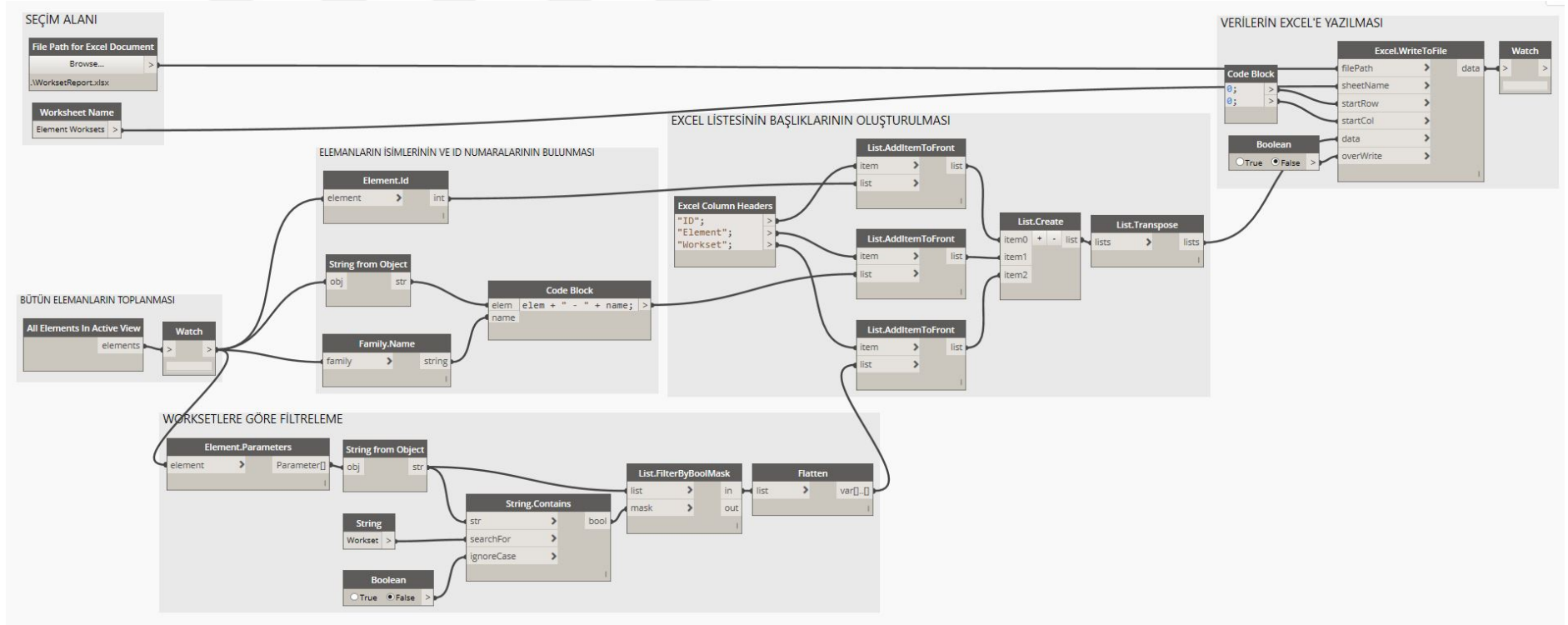
Şekil 4.109 : Dynamo ile hazırlanan algoritma

Nesnelerin hangi worksetlere ait olduğunun rapor edilmesi

Yapı bilgi modelinde yer alan nesnelerin hangi worksetlerde olduğunu öğrenmek ve elde edilen verileri Excel tablosuna ihraç etmek için Dynamo programı ile bir algoritma hazırlanmıştır. Algoritmanın yürütülmesi sonucunda elde edilen verilen hangi Excel dosyasına yazdırılacağına öncelikle olarak tanımlanması gerekmektedir. Revit’te yer alan bütün nesneler seçilir ve eleman ID’sine, ismine ve worksetine göre Excel tablosunda belirtilen sütuna yazılır (Şekil 4.11). Algoritmanın yürütülmesi sonucu Şekil 4.110’da gösterildiği gibi bir Excel tablosu elde edilir.

	A	B	C
1	ID	Element	Workset
2	201028	Floor - Generic 150mm	Workset : Floors
3	201030	Wall - Exterior - Brick on Mtl. Stud	Workset : Walls
4	201031	Wall - Exterior - Brick on Mtl. Stud	Workset : Walls
5	201032	Wall - Interior - Partition	Workset : Walls
6	201033	Wall - Interior - Partition	Workset : Walls
7	201034	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
8	201036	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
9	201038	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
10	201040	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
11	201042	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
12	201044	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
13	201046	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
14	201048	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
15	201050	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
16	201052	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
17	201054	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
18	201056	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
19	201058	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
20	201060	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
21	201062	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
22	201064	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
23	201066	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
24	201068	Family=W14X90, Type=W14X90 - W14X90	Workset : Workset1
25	201070	Window - Window	Workset : Workset1
26	201071	Wall - Interior - Partition	Workset : Walls
27	201072	Wall - Gallery Wall	Workset : Walls
28	201073	Curtain Panel - TYP	Workset : Walls

Şekil 4.110 : Dynamo ile Excel’e yazılan veriler



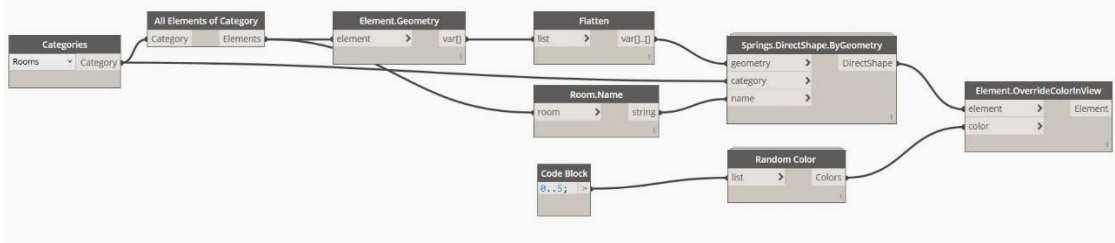
Şekil 4.111 : Dynamo ile hazırlanan nesnelerin hangi worksette olduğunun raporlanması algoritması

4.2.2.3 Dynamo ile modeldeki verilerin görselleştirilmesi

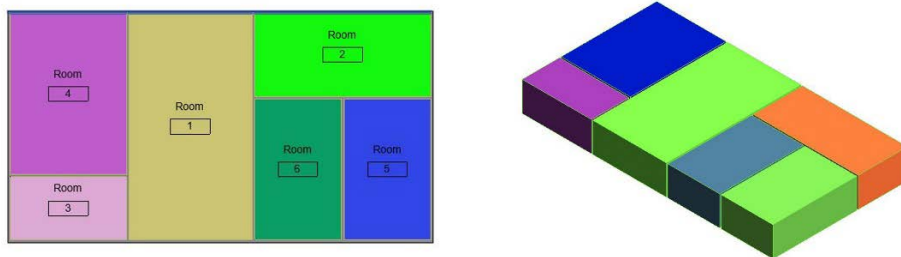
Veri görselleştirme, veri tabanlarında bilgi bulma ile ilgili görevler için kullanılan önemli bir yaklaşımdır. Çoğunlukla görselleştirilecek veriler karmaşıktır, birden fazla boyuta veya özelliğe sahip ve pek çok bireysel veri noktasından oluşup, geleneksel piksel tabanlı ve geometrik tekniklerle görselleştirmeyi zorlaştırmaktadır. Dynamo, veri görselleştirmesi ile yapı bilgi modelinde yer alan karmaşık verilerin anlaşılmasını kolaylaştırmaktadır. Görselleştirme genellikle verilerin görsel bir temsil ile eşlenmesi olarak tanımlanabilmektedir. Görsel temsiller, iyi yapılandırılmışlarsa, bilgiyi sadece kullanıcılara hızlı bir şekilde sunmak için değil, aynı zamanda veri kavrayışını desteklemek ve en üst düzeye çıkarmak için de yararlı olabilmektedir.

Odaların görselleştirilmesi

Yapı bilgi modelinde yer alan odaların görselleştirilmesi Dynamo ile birlikte yapılabilmektedir. Dynamo ile hazırlanan algorithmada öncelikle modelde yer alan odalar seçilir. Daha sonra Springs paketine ait “Springs.DirectShape.ByGeometry” düğümü ile odaların görselleştirilmesi sağlanır (Şekil 4.112). Görselleştirme sırasında kullanılan renkler ise rastgele seçilebilmektedir. Görselleştirilen oda hacimleri, Dynamo programının geometri ön izlemesi ile ve Revit programında yer alan 2D ve 3D görüntülerinden görüntülenebilmektedir (Şekil 4.113)



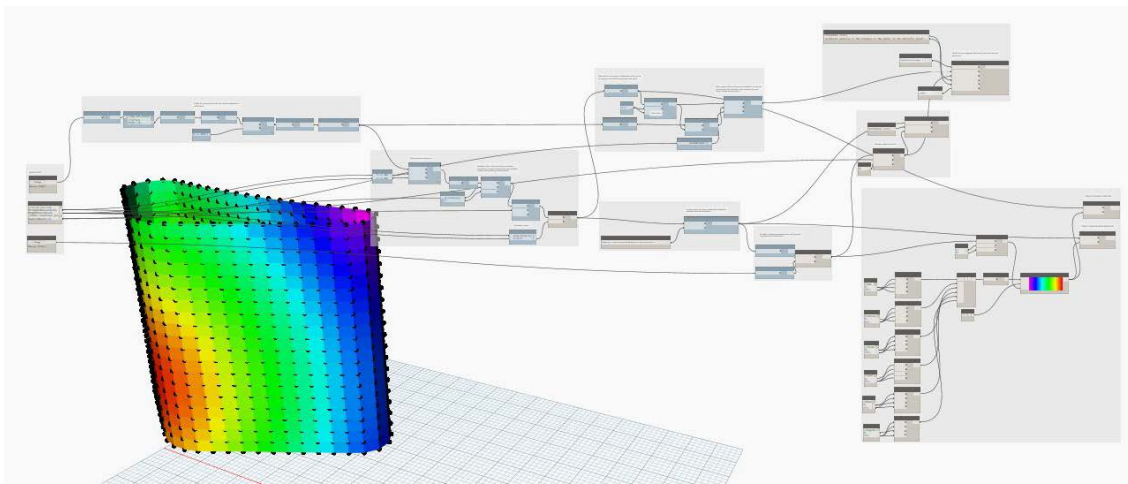
Şekil 4.112 : Dynamo ile odaların görselleştirilmesi algoritması



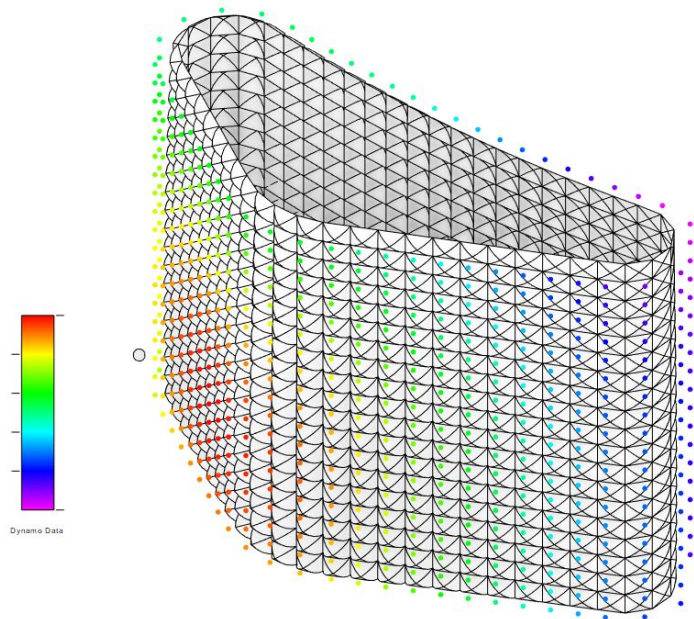
Şekil 4.113 : Odaların görselleştirilmesi

Cephe tasarımı ve analizi

Dynamo programında hazırlanan algoritma ile Revit programında tanımlanan kütle(mass) seçilerek yüzey üzerine uyarlanabilir cephe panelleri (adaptive components) yerleştirilmiştir. Yerleştirilen panellerin belli bir noktaya olan uzaklığı analiz edilmiş ve renklerle görselleştirilmesi yapılmıştır (Şekil 4.112). Dynamo aracı ile yapılan görselleştirme Revit ortamında da görüntülenebilmektedir (Şekil 4.113). Görselleştirme amaçları, öğrenilmek istenilen bilgi türüne bağlı olarak çeşitlendirilebilmektedir.



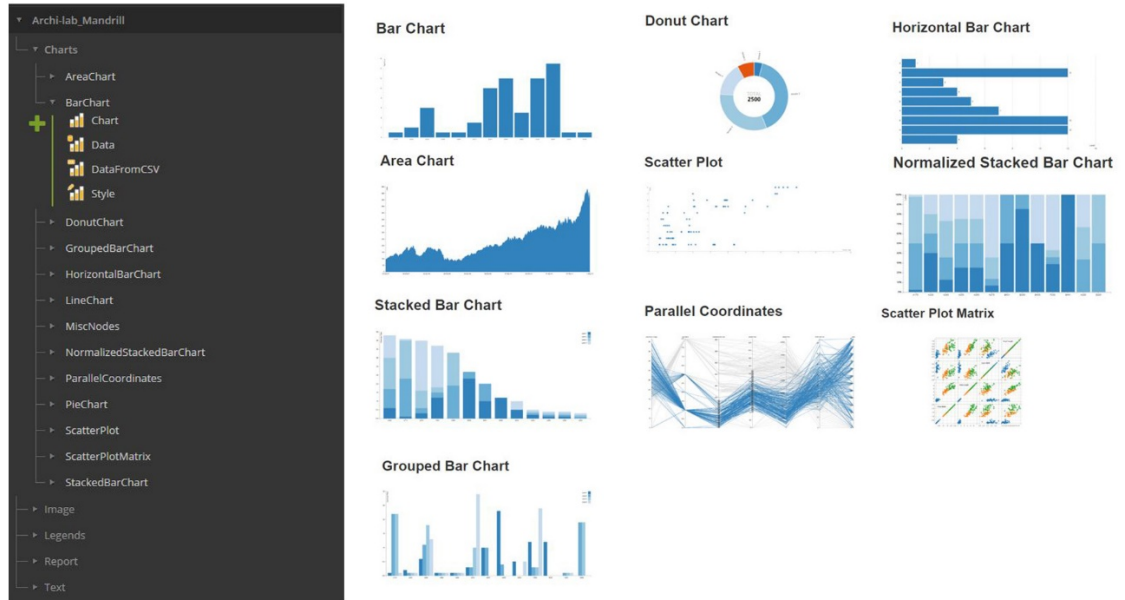
Şekil 4.114 : Dynamo ile hazırlanan algoritma ve görselleştirilen cephe



Şekil 4.115 : Revit programında, uyarlanabilir bileşenlerin görüntülenmesi

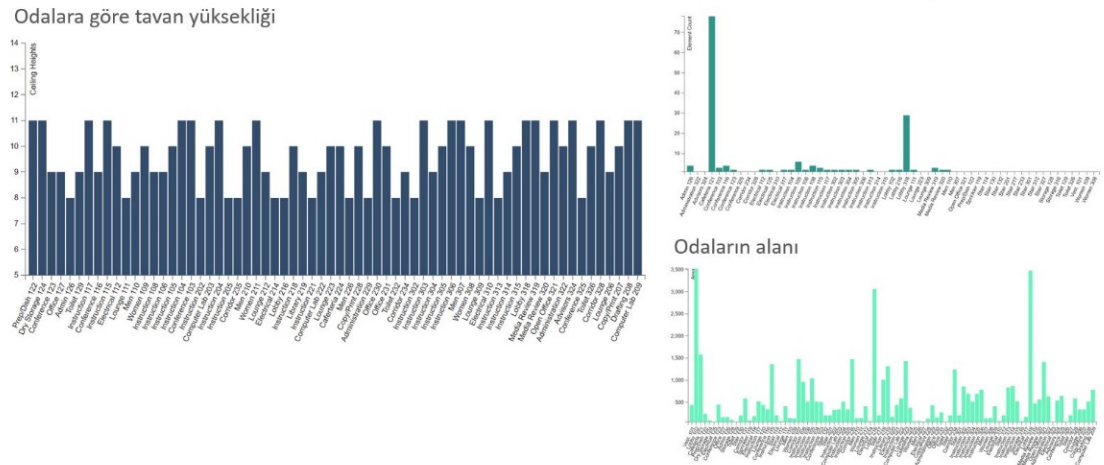
Mandrill paketi ile verilen görselleştirilmesi

Mandrill, veri görselleştirme amacıyla geliştirilen bir Dynamo paketidir. Archi-lab paketinin geliştiricisi Konrad Sobon tarafından geliştirilmiştir. Mandrill verileri birçok farklı gösterim şekliyle temsil edilebilmektedir (Şekil 4.116). Mandrill ile veri görselleştirmesine örnek olarak oda bilgilerinin görselleştirilmesi verilebilir (Şekil 4.117). Dynamo aracı ile hazırlanan algoritma sonucu görselleştirilen veriler Dynamo ortamında bir pencere yardımıyla görüntülenebilirken, PDF ya da HTML formatında dışarıya aktarılabilmektedir.



Şekil 4.116 : Mandrill paketi ve veri görselleştirme grafikleri (Mandrill Primer, t.y.)

Oda Bilgilerinin Görselleştirilmesi

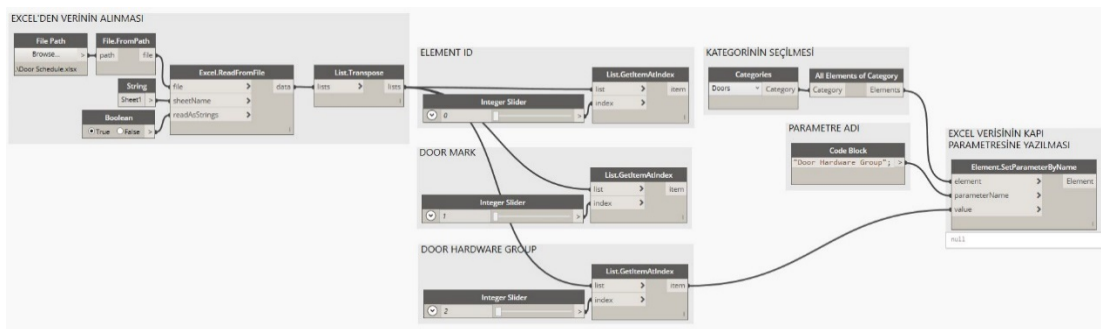


Şekil 4.117 : Mandrill paketi ile oda bilgilerinin görselleştirilmesi

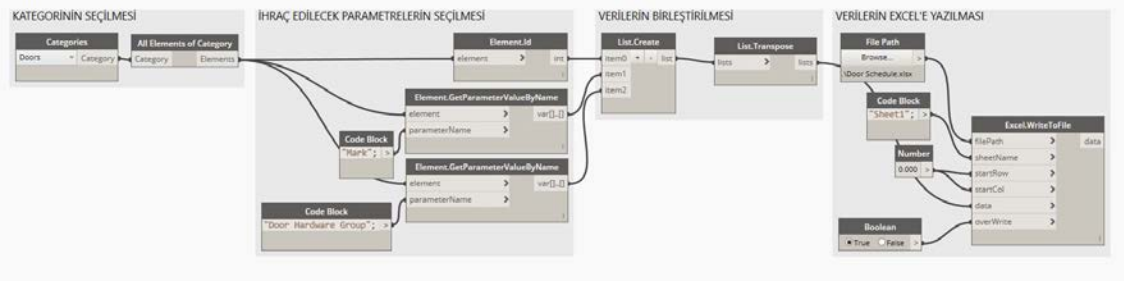
4.2.3 Harici yazılım ve donanım entegrasyonu

Dijital dünyayı yapıyı çevremize bağlamak, yeni fikirler geliştirmek ve çevremizi daha iyi anlamak için heyecan verici bir yol olabilmektedir. Arduino gibi ucuz programlanabilir cihazların ortaya çıkmasıyla bu durum her zamankinden daha kolay olmaktadır (Tomasetti, t.y.). Dynamo ile Kinect veya Arduino ile bağlantı kurarak sensör verilerine erişmek mümkündür. Endüstriyel robotların Dynamo'ya bağlanmasıyla modeldeki nesnelerin bilgisayar kontrollü imalatı, kitlesel kişiselleştirmeye doğru atılan adımlardan sayılabilmektedir. Dynamo'ya bağlantı içeren yazılım listesi sürekli olarak artmaktadır. Birlikte çalışabilirliği genişletmek için Dynamo ile etkileşimde olan yazılım ve donanımlara örnek olarak Excel, Rhino, Navisworks, Robot Structural Analysis ve SQL verilebilmektedir.

Yapı bilgi modellerinin önemi veri zengini olmalarıdır. Mevcut YBM programları ile bu veriler, model içerisine alınır veya modelden ihraç edilebilir. Revit programı ile istenilen çizelgeler oluşturulup, Excel ortamına aktarılabilmektedir. Revit bu yetenekleri geliştirmek için adımlar atmış olsa da yürütülen bu süreç birçok iş adımı gerektirirken zaman ve emek kaybına neden olmaktadır. Dynamo kullanılarak Revit modeli ile Excel arasında iki yönlü bir bağlantı oluşturulabilmektedir. Öncelikle, belirli verileri Excel'e aktarmak için bir Dynamo algoritması oluşturmak gerekmektedir. Veriler Excel ile gerektiği gibi değiştirildikten sonra verilerin modele alınması için yine Dynamo aracından yararlanılmaktadır. Şekil 4.118'de Excel dosyasında hazırlanan kapılara ait parametreler Dynamo ile içeriye aktarılmış ve ilgili veriler, Revit'te yer alan kapı parametrelerine yazılmıştır. Şekil 4.119'da ise Revit verilerinin Excel ortamına ihraç edilmesi için bir algoritma hazırlanmıştır.



Şekil 4.118 : Excel'den Revit programına veri alınması

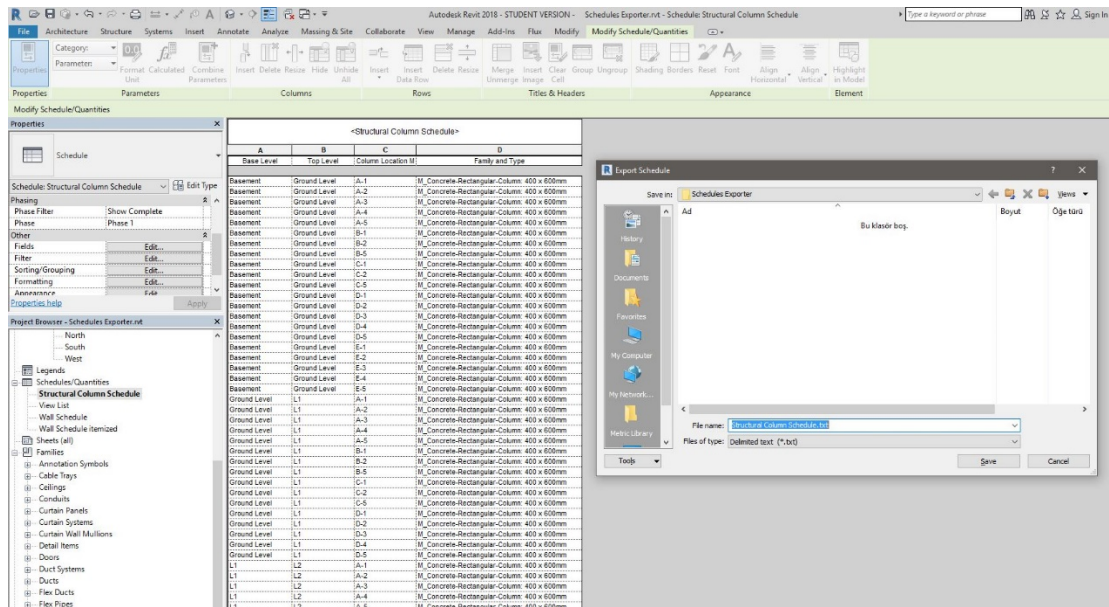


Şekil 4.119 : Revit programından Excel'e veri yazılması

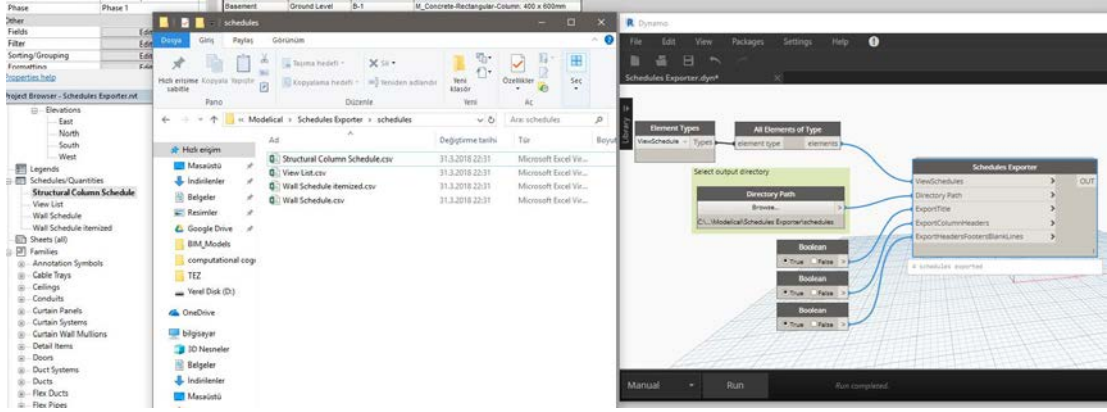
Revit'te oluşturulan çizelgelerin Excel'e ihraç edilmesi

Dynamo ile birlikte Revit'te hazırlanan çizelgelerin ihraç edilmesi örneğine geçmeden önce kısaca bu işlemin sadece Revit yazılımı ile nasıl gerçekleştirildiğini anlatmak karşılaştırma yapmak açısından önemlidir.

Revit ortamında hazırlanan çizelgeler ayrı ayrı yazı formatında ihraç (export) edilir (Şekil 4.120). Yazı formatında elde edilen çizelgeler Excel ortamına ayrı ayrı aktarılır ve gerekli düzenlemeler yapılır (Şekil 4.121). Bu süreçte birbirini tekrar eden iş adımları görülmektedir. Bu sürecin zaman aldığı ve bu tarz işlerin mimarın, mühendislerin tasarım konularında yoğunlaşması gereken işlere daha az zaman ayırmasına neden olduğu görülmektedir.



Şekil 4.120 : Revit ile çizelgelerin yazı formatında ihraç edilmesi



Şekil 4.124 : Çizelgelerin Dynamo ile birlikte Excel formatında ihraç edilmesi

Revit programında çizelgelerin ayrı ayrı ihraç edilmesi sonucunda oluşan zaman kaybının Dynamo’da hazırlanan algoritma ile önüne geçilebilmektedir. Hazırlanan algoritmanın çalıştırılması sonucu Revit programında yer alan bütün çizelgeler aynı anda Excel formatında ihraç edilebilmektedir. Şekil 4.124’de gösterildiği gibi Revit’te hazırlanan 4 çizelge aynı anda 4 Excel formatına dönüştürülmüştür. Dynamo ile hazırlanan bu algoritma her proje için Dynamo Player ile birlikte yürütülebilmektedir. Böylece çizelge ihraç edilmesi sırasında birbirini tekrarlayan iş adımlarının önüne geçilebilmektedir.

4.3 Bölüm Değerlendirmesi

Görsel programlama ile ilgili literatür araştırması, ilk görsel programlama araçlarının başlangıcındaki hedeflerin günümüzdeki hedeflere benzer olduğunu açıkça göstermektedir. Bu hedeflerden biri, mimari tasarım iş akışlarının verimliliğini sağlamaktır. Son on yılda MMİ endüstrisindeki tasarım süreci, YBM tabanlı araçlar ile büyük ölçüde etkilenmiştir. Araştırma ile tasarımcılar ve mühendisler için uygun hesaplamalı tasarım yaklaşımlarının eksikliği tespit edilmiştir. Grasshopper aracının başarısı, programcı olmayan kullanıcılar tarafından kullanılabilen özellikle geometri anlamında kullanıcıya oldukça esneklik sağlayan bir araç olmasıdır. Ancak, Grasshopper mevcut YBM yazılımları ile entegre bir şekilde çalışmadığı ve yapı bilgi modelinde yer alan verilere müdahale edemediği için sektörde yeni bir kavram ortaya çıkmıştır: YBM tabanlı hesaplamalı tasarım. Bu bağlamda, YBM tabanlı hesaplamalı tasarım aracı olan Dynamo ve kullanım alanları uygulamalar ile birlikte ayrıntılı olarak incelenmiştir.

Mevcut YBM ile yürütülen çalışmalar sırasında harcanan emek ve zamanın hesaplamalı tasarım aracı olan Dynamo ile birlikte nasıl minimize edilebileceği ile ilgili yeni iş akışlarının oluşturulması mümkündür. Tez kapsamında mimari tasarım sürecinin erken tasarım aşamasından teslim sürecine kadar olan proje süreçlerinde MMİ sektöründeki kullanıcıların karşılaştığı zorlukların ve zaman kayıplarının hangi görevlerde meydana geldiği, literatür araştırması ve ofis deneyimlerinden edinilen bilgiyle ortaya konulmuştur. Erken tasarım aşamasında kullanıcı mevcut YBM sistemlerine Dynamo aracını entegre ederek üretken tasarım yöntemleri ile birçok tasarım alternatifine erişebilmektedir. Erken tasarım aşamasında kullanıcı, veri odaklı tasarım kararlarını Dynamo’da yapılan analiz ve optimizasyonlarla birlikte verebilmektedir. Erken tasarım aşamasında analiz ve optimizasyonlara göre verilen kararlar ile projenin ilerleyen aşamalarında yapılacak olan karar değişikliklerinin sayısı azaltılırken, karar değişikliklerine bağlı olarak artan proje maliyetlerinin de düşürülmesi hedeflenmektedir. Projenin erken tasarım aşamasından uygulama aşamasına kadar olan süreçte projenin büyüklüğü, modelin olgunluk seviyesi ile artan verilerin miktarı, projenin detay seviyesi, paydaşların sayısı gibi faktörler ile modelin yönetimi zorlaşmaktadır. Proje revizyonları ile birlikte birbirini tekrar eden işlerin sayısı artmaktadır. Revizyon işlerinin yapılması sırasında kullanıcıya bağlı olarak yapılan hata sayısı da artmaktadır. Sonuç olarak, hesaplamalı tasarım aracı olan Dynamo ile projenin uygulama aşamasında modelde artan veri ile birlikte oluşan zaman kaybı ve iş yükünün azaltılması hedeflenmektedir. Ayrıca Dynamo ile modelde yer alan verilerin yönetilmesi ile proje süreçlerinde yaşanan veri kaybının önüne geçilebilmektedir.

Tez kapsamında Dynamo ile birlikte birçok özel probleme cevap olarak üretilmiş olan Dynamo paketleri, YBM kavramsal tasarım aracı olan Formit aracı, bulut sistemleri, Power BI gibi veri görselleştirme programları ve MMİ sektöründe oldukça kullanılan Excel gibi diğer araçlardan yararlanılmaktadır. Bu araçların projenin farklı aşamalarında kullanıcının ihtiyacına yönelik olarak Dynamo ile birlikte iş akışlarını nasıl optimize edebileceği uygulamalar ile birlikte ortaya konulmaktadır. Oluşturulan yaklaşım yöntemlerinin, YBM sistemlerini kullanan ve proje adımlarını optimize etmek isteyen ofisler için örnek teşkil edebileceği düşünülmektedir. Dynamo’nun belirli standartlarının olmaması ofislerin YBM tabanlı hesaplamalı tasarım yaklaşımı

benimsemelerinin önündeki engellerden birisi olabilmektedir. Ofisler, kendi Dynamo standartlarını oluşturarak bu durumun önüne geçebilmektedirler.

Dynamo açık kaynak kodlu olduğu için sürekli geliştirilmektedir. Bu durumun sektörde karşılaşılan problemlere özel olarak birçok çözümü beraberinde getirmenin yanı sıra birtakım olumsuzluklara sebep olduğu görülmektedir. Dynamo'nun hızlı gelişimi ile birlikte özel düğümlerden bazıları zamanla güncellenebilmektedir. Bu durum Dynamo'yu yeni öğrenmeye başlayan kullanıcılar için zorlayıcı olabilmektedir. Fakat Dynamo'nun gelişimine bağlı olarak pek çok sanal kitaplar ve videolar yayınlanmaktadır. Autodesk University bünyesinde Dynamo ile ilgili yayınlanan dersler yıllara göre artmaktadır. Bu kaynaklarla birlikte Dynamo kullanıcısı, Dynamo'nun gelişimine bağlı olarak kendi bilgisini güncelleyebilmektedir.

Tezin yazıldığı dönem içerisinde Dynamo geliştirilmeye devam eden bir program olduğu için akademik araştırmalardan çok programın geliştiricileri tarafından yayınlanmış sanal kaynaklardan yararlanılmıştır.

Sonuç olarak, Dynamo'nun görsel programlama diline sahip olması, YBM tabanlı olması, mimarlık, mühendislik ve inşaat sektöründe yer alan ofislerde iş akışlarını optimize etmek için kullanılmak üzere geliştirilmesi, eklentiler ile desteklenebilmesi gibi özellikler kullanıcılar için oldukça faydalıdır. Kullanıcılar, Dynamo ile kavramsal tasarım aşamasında daha çok analiz ve optimizasyonlar yaparken projenin uygulama aşamasında yapı bilgi modelinde yer alan verinin yönetimi, veri manipülasyonu, veri aktarımı ve belgeleme çalışmaları ile ilgili görevleri yapabilmektedirler.



5. SONUÇ VE ÖNERİLER

Mimari tasarım süreci gelişen teknoloji sistemleri ile giderek derinleşen ilişkiler içerisinde. Geçmişte tasarımcılar projelerini oluşturmak ve temsil etmek için tek araç olarak hayal gücü ve çizgileri kullanmaktaydılar. Bu durum mimari tasarım kararlarının çoğunun tasarımcının sezgilerine ve tecrübelerine dayandırılmasına sebep olmaktaydı. Bununla birlikte tasarımcının zihninde üç boyutlu olarak görselleştirilebilen tasarımların süreç içerisinde farklı disiplinlerle paylaşılması ve anlaşılması gerekmektedir. Bu durum tasarımların temsil meselesini ortaya çıkarmıştır.

Yapı endüstrisi dijitalleşmeye başladığında, geleneksel çizim masası iki boyutlu bir bilgisayar destekli tasarım (Computer Aided Design-CAD) sistemine dönüşmüştür. Bilgisayarların yaygınlaşması ve CAD yazılımlarının ortaya çıkması ile tasarım sürecinde sayısal araçların kullanımı başlamıştır. Daha sonra 2D CAD sistemine ek olarak 3D özelliği eklenmiştir.

Günümüzde bilgisayarlar evrim geçirmekte ve mimarlık, mühendislik ve inşaat sektörü de bilgi çağının dönüşümü olarak paradigma değişimi yaşamaktadır. Artık, bilgisayarlar ve yazılımlar büyük miktarda veriyi işleyebilmektedir. Bununla birlikte yapı endüstrisinde CAD sistemlerinin yerini YBM sistemleri almaya başlamıştır. Böylelikle yapı endüstrisi yeni bir dönüşümle karşılaşmış, buna bağlı olarak yeni iş akışları ve roller ortaya çıkmıştır. Mimarlık, mühendislik ve inşaat sektöründe yaşanan bu dönüşüm uluslararası alanda hızla yaygınlaşmış ve bazı ülkelerde YBM sisteminin özellikle kamu projelerinde kullanılması zorunlu hale getirilmiştir.

YBM'nin ortaya çıkışı ve gelişim sürecine paralel olarak hesaplamalı tasarım yaklaşımı da sektörde kendine geniş bir kullanım alanı bulmuştur. Hesaplamalı tasarım ve YBM sistemleri farklı ortamlarda geliştirilmeye devam ederken sektörde meydana gelen ihtiyaçlar doğrultusunda ortak bir platformda buluşturulmuştur.

YBM sisteminin en önemli özelliği kullanıcılara veri-zengini bir ortam sunmasıdır. YBM sisteminin olmazsa olmazı verilerin modele eklenmesi, sistem içerisinde

işlenmesi, düzenlenmesi ve gerekli disiplinler tarafından istenildiği zaman modelden alınabilmesidir. Mevcut YBM yazılımları verileri sisteme işleme, manipüle ve optimize etme gibi konularda sınırlı yetenekler sağlamaktadır. YBM modellerinde veri yükünün artması ile tasarım kararlarında alınan değişikliklere karşı modelin kolay adapte olamaması, katmanlaşan veri tabanından istenilen verilerin alınmasının zorlaşması, kullanıcının bu süreci yönetme konusunda yetersiz kalması sonucu hata sayısının artması ile birlikte YBM ve hesaplamalı tasarım ilişkisi oldukça ilgi görmektedir. Bu duruma cevap olarak, YBM tabanlı hesaplamalı tasarım aracı olan Dynamo ortaya çıkmıştır. Dynamo ile Grasshopper aracı güçlü bir benzerlik gösterebilir. Her iki hesaplamalı tasarım aracı da görsel programlama dilini kullanarak algoritmaların üretilmesine ve tasarımın manipüle edilmesine izin vermektedir. Grasshopper çoğunlukla tasarım morfolojisini etkileyen geometrik ilişkileri yönetmek için kullanılırken Dynamo, kullanıcılara proje verilerini manipüle etmek için erişim sağlamaktadır. Böylece Dynamo, Grasshopper aracını kopyalamaktan çok daha fazlasını yapmaktadır.

Dynamo aracı, YBM'nin kendine özgü bir veri tabanı oluşturmaya ve bu verileri belirli amaçlar doğrultusunda yönetmesine olanak tanıdığı için kullanımı giderek artmaktadır. Dynamo, YBM verilerini işleme, yönetme ve bu verilerin dış kaynaklarla etkileşime girebilmesi konusunda birçok olanak sağlamaktadır. Böylece modelleme ve veri yönetimi süreçleri arasında güçlü bir birliktelik oluşturulabilmektedir.

Mimarlık, mühendislik ve inşaat sektöründe YBM tabanlı mimari tasarım süreçlerinde kullanılan mevcut yöntemlerin zaman ve maliyet kaybına sebep olduğu görülürken iş akışlarının verimliliğinin sağlanması ihtiyacı doğmuştur. Bu gereksinimden yola çıkılarak tez kapsamında YBM tabanlı mimari tasarım süreçlerinde iş gücü ve zaman kaybına sebep olan, verimliliğin sağlanması gereken iş akışlarının neler olduğu araştırılmaktadır. Ayrıca mimari tasarım sürecinde tasarım kararlarını güvenilir bir şekilde verebilmek, sürdürülebilir yapılar inşa edebilmek adına yapılması gereken analiz ve optimizasyonlar Dynamo ile birlikte yürütülebilmektedir. Bu bağlamda, YBM mimari tasarım süreci ile hesaplamalı tasarım yaklaşımının entegre olması ile birlikte elde edilen yeni iş akışları ile sürecin verimliliğinin sağlanabileceği üzerinden tez çalışması yürütülmektedir.

Tez kapsamında, hesaplamalı tasarım yaklaşımı olan görsel programlama aracı Dynamo'nun kullanılmasının, YBM tabanlı mimari tasarım süreçlerine olan etkileri ile ilgili yapılan uygulamalar ve araştırmalarla test edilmesi sonucu aşağıdaki bulgulara varılmaktadır.

- YBM tabanlı mimari tasarım süreçlerinde Dynamo'nun kullanılması ile yeni iş akışları oluşturulmaktadır.
- YBM tabanlı mimari tasarım süreçlerinde iş gücü ve zaman kaybının fazla olduğu işler Dynamo aracı ile optimize edilebilmektedir. Tekrara dayalı olarak yapılması gereken işlerin algoritmalarla çözülebilmesi proje sürecinin verimliliğini artırmaktadır.
- Yapı bilgi modelinin olmazsa olmazı olan veri ile ilgili aktarma, işleme, düzenleme, manipüle etme ve disiplinlerle paylaşımı gibi konularda Dynamo birçok olanak sağlamaktadır.
- Ofislerin, YBM tabanlı mimari tasarım süreci içerisinde karşılaştıkları problemleri Dynamo ile optimize edebilmeleri için ofis içi belirli standartların oluşturulması gerekmektedir. Ofis içerisinde farklı seviyelere sahip kullanıcılar tarafından Dynamo'da yazılan algoritmanın anlaşılabilir olması ve sorunsuz bir şekilde yürütülebilir olması önemli olmaktadır.
- Ofisler, mimari tasarım süreçleri içerisinde optimize edilmeye gerekli görülen işleri tespit ederek ilgili çözüm algoritmalarını üretebilmektedirler. Bu bağlamda Dynamo ile süreçlerin optimizasyonunun daha verimli ilerleyebilmesi adına hazırlanan algoritmalar birçok projede denenerek ofisin Dynamo kütüphanesi oluşturulabilmektedir. Böylece tek bir proje sürecini optimize etmenin ötesine geçilebilmektedir.
- Dynamo ile mimari tasarım süreçlerinin erken tasarım aşamasında tasarım ile ilgili gerekli analiz ve optimizasyonlar yapılabilir. Yapı için daha iyi bir performans elde edebilmenin yolu, analizlerin geri besleme döngüsü ile çalışabilmesidir. Böylece analizler tekrarlandığında yapılan yinelemelerin tasarımı nasıl etkilediği konusunda hızla geri bildirim alınabilmektedir. Dynamo ile yapılan analizler doğrudan YBM yazılımı ile entegre bir şekilde gerçekleştirilebildiğinden dolayı proje süreci içerisinde farklı yazılımlar arasında dosya paylaşımına bağlı olarak görülen veri kayıpları

görülmemektedir. Analiz ve optimizasyon uygulamaları için Dynamo aracına birçok farklı araç entegre olabilmektedir: bulut tabanlı çalışan Project Fractal gibi. Bu eklentiler ile Dynamo'nun olanakları daha da genişletilmektedir.

- Mimari tasarım süreci içerisinde gerçekleştirilen analiz ve optimizasyon uygulamaları için gerekli olan dış kaynak verileri Dynamo aracı ile kolaylıkla sisteme aktarılabilir. Örneğin; gün ışığı, ses kaynağı gibi analiz süreci için gerekli olan dış kaynaklı veriler proje sürecine eklenebilmektedir. Bu araçların kullanılması ile mimari tasarım süreci için önemli olan veri odaklı tasarımın gerçekleştirilmesi mümkün olmaktadır.
- Dynamo aracı ile YBM yazılımlarına gömülü olan veriler görselleştirilebilmektedir. Veri görselleştirme, karmaşık bilgileri, hedefleri ve temsilleri tasarım konusunda profesyonel olmayan kişilere iletmeye yardımcı olmaktadır. Tasarımcıların proje ile ilgili aktarmak istediklerinin görsel bir şekilde karşı tarafa aktarılmasına yardımcı olmaktadır.
- Dynamo'nun görsel programlama diline sahip olması programcı olmayan kullanıcılar tarafından kullanılması için motive edici olmaktadır.
- Dynamo aracının açık kaynak kodlu olması ve geliştiricilerinin sektördeki problemler ile yakından ilgilenmeleri sebebi ile birçok özel paket ve düğümler oluşturulmaktadır. Dynamo'nun geliştirilmesi ile YBM tabanlı mimari tasarım süreçlerinde görülen problemlerin özel düğümlerle çözülebilmesi Dynamo'nun kullanımına olan ilgiyi artırmaktadır.
- Geleneksel yöntemlere göre kullanılan hesaplamalı tasarım araçları ile mimarın çalışmasında yaşanan değişim fazladan bir yük olarak görülebilse de aslında mimarın kendi ihtiyaçlarına göre uygun araçlar tasarlamasına ve belirli görevleri gerçekleştirmek için yazılımı istediği şekilde kullanabilmesine olanak tanıdığı için birçok potansiyel barındırmaktadır. Dynamo ortamında özel düğümler geliştirerek ihtiyaca yönelik algoritmalar üretmek oldukça kolay olmaktadır. Bununla birlikte mevcut araçları programlamak ve geliştirmek, yalnızca bir dil meselesi olmamaktadır. Aynı zamanda süreç içerisinde karşılaşılan problemlerin yapılandırılması ve bununla mücadele etme yöntemidir. Görsel programlama ile algoritmaların oluşturulması mantıksal yöntemler geliştirirken yeni bir dil öğrenmek zorunda kalınmadan yazılımın

anlaşılmasını sağlamaktadır. Böylece Dynamo, tasarımcıların düşüncelerini ve çalışma süreçlerini yapılandırmalarına yardımcı olmaktadır.

Sonuç olarak, tez kapsamında Dynamo'nun mevcut YBM tabanlı mimari tasarım iş akışlarının verimliliğinin nasıl sağlanabileceği araştırılmıştır. Bu bağlamda yapılan literatür araştırması, araştırmacının kendi ofis deneyimleri ve uygulanan örneklerle birlikte hesaplamalı tasarım yaklaşımları ile mevcut YBM süreçlerinde görülen zaman ve gereksiz iş gücü kayıplarının önüne geçebildiği gözlemlenmektedir. Ayrıca, hesaplamalı tasarım yaklaşımı ile birlikte veri yönetiminin beraberinde getirdiği doğru model üretiminin başarısı proje sürecinin her evresine yansımaktadır. Tez kapsamında yapılan uygulamalar ve ortaya konulan bulguların YBM tabanlı mimari tasarım süreçlerini kullanan ofisler için faydalı olabileceği düşünülmektedir.



KAYNAKLAR

- Aamodt, A. ve Nygård, M.** (1995). Different roles and mutual dependencies of data, information, and knowledge-an AI perspective on their integration. *Data ve Knowledge Engineering*, 16, ss. 191-222.
- Ackoff, R.L.** (1990). From data to wisdom. *Informatie*, 32, ss. 486-90.
- AIA, The American Institute of Architects.** (2007). Integrated project delivery: a guide. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <https://www.aia.org/>
- Aish, R., Fisher, A., Joyce, S. ve Marsh, A.** (2012). Progress towards Multi-Criteria Design Optimisation using DesignScript with SMART Form, Robot Structural analysis and Ecotect building performance analysis, ACADIA. Erişim adresi: http://companyshed.com/downloads/documents/2012_ACADIA.pdf
- Autodesk, Project Fractal.** (t.y.). Erişim tarihi: 28 Nisan 2018, Erişim adresi: <https://www.fractal.live/share/576847113a47345a0407ee56>
- Asl, M. R., Bergin, M., Menter, A., Yan, W.** (2014). BIM-based parametric building energy performance multi-objective optimization, The 32nd International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe 32), UK: Northumbria University, ss. 455–464.
- Asl, M. R., Stoupine, A., Zarrinmehr, S. ve Yan, W.** (2015a). ‘Optimo: A BIM-based Multi-Objective Optimization Tool Utilizing Visual Programming for High Performance Building Design’, eCAADe 33, ss. 673–682. Erişim adresi: https://www.researchgate.net/profile/Mohammad_Rahmani_Asl/publication/283351762_Optimo_A_BIM-based_Multi-Objective_Optimization_Tool_Utilizing_Visual_Programming_for_High_Performance_Building_Design/links/5636806508aeb786b703f21d.pdf
- Asl, M. R., Zarrinmehr, S., Bergin, M., ve Yan, W.** (2015b). ‘BPOpt: A framework for BIM-based performance optimization’, *Energy and Buildings* 108, ss. 401–412. Erişim adresi: <https://www.sciencedirect.com/science/article/pii/S0378778815302528?via%3Dihub>
- Barnes, P. ve Davies, N.** (2015). BIM in Principle and Practice, Institute of Civil Engineers.

- Bender, S. ve Fish, A.** (2000). The transfer of Knowledge and the Expertise: the continuing need for Global Assignments. *Journal of Knowledge Management*, Vol. 4, No. 2, pp. 125-137
- Bew, M. ve Richards, M.** (2008). Bew-Richards BIM maturity model, BuildingSMART Construct IT Autumn Members Meeting, Brighton.
- Björk, B.C.** (1995). Requirements and Information Structures for Building Product Models. Helsinki, VTT Technical Research Center of Finland.
- Blackwell, A. ve Rodden, K.** (2003). Preface to this Electronic Edition, Sketchpad: A man-machine graphical communication system, ss. 3–5. Erişim adresi: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>
- Boeykens, S.** (2012). Bridging building information modeling and parametric design, eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012.
- Braumann, J. ve Brell-Cokcan, S.** (2014). Visual robot programming: linking design, simulation, and fabrication, Proceedings of the Symposium on Simulation for Architecture ve Urban Design. Tampa, Florida: Society for Computer Simulation International, ss. 1–8.
- Brodth, W.** (2016). Omniclass. Erişim tarihi: 28 Nisan 2018, Erişim adresi: WBDG: <http://www.wbdg.org/resources/omniclass.php>
- Celani, G. ve Vaz, C. E. V.** (2012). CAD scripting and visual programming languages for implementing computational design concepts: A comparison from a pedagogical point of view, *International Journal of Architectural Computing*, 10(1), ss. 121–138.
- Charette, R. P. ve Marshall, H. E.** (1999). UniFormat II elemental classification for building specifications, cost estimating and cost analysis. NIST.
- Coenders, J. L.** (2009). Parametric and associative design as a strategy for conceptual design and delivery to BIM, Symposium of the International Association for Shell and Spatial Structures, Valencia. Erişim adresi: https://riunet.upv.es/bitstream/handle/10251/6981/PAP_COENDERS_1112.pdf?sequence=1&isAllowed=y
- Cormen, T. H.** (2009). Introduction to algorithms. 3rd ed. Cambridge, Mass: MIT Press.
- CSI and CSC.** (2004). MasterFormat 2004 edition numbers ve titles. The Construction Specifications Institute (CSI) and Construction Specifications Canada (CSC).
- Czmoch, I. ve Pękala, A.** (2014). Traditional Design versus BIM Based Design. *Procedia Engineering*, 91, ss. 210-215.

- Das, S., Day, C., Hauck, J., Haymaker, J. ve Davis, D.** (2016). Space Plan Generator: Rapid Generation ve Evaluation of Floor Plan Design Options to Inform Decision Making. ACADIA // 2016: POSTHUMAN FRONTIERS: Data, Designers, and Cognitive Machines [Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), ss. 106-115. Eriřim adresi: http://papers.cumincad.org/cgi-bin/works/Show?acadia16_106
- Davenport, T. H. ve Prusak, L.** (2000). Working knowledge: how organizations manage what they know. Harvard Business Press, 1998.
- Deutsch, R.** (2015). Data-Driven Design and Construction: 25 strategies for capturing, analyzing and applying building data. Hoboken: John Wiley ve Sons Inc.
- Dino, İ. G.** (2012). Creative Design Exploration by Parametric Generative Systems in Srchitecture, METU Journal of Faculty of Architecture, 29(1), ss. 207–224.
- Dynamo Primer.** (t.y). Eriřim tarihi: 28 Nisan 2018, Eriřim adresi: <http://dynamoprimer.com/en/>
- Eastman, C., Teichholz, P., Sacks, R., ve Liston, K.** (2011). BIM Handbook- A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, 2nd ed., New Jersey, John Wiley and Sons.
- Ekholm, A.** (1996). A conceptual framework for classification of construction works. ITcon (Electronic Journal of Information Technology in Construction) vol.1, p.1-25.
- Ekholm, A. ve Häggström, L.** (2011). Building classification for BIM – Reconsidering the framework. Proceedings of the CIB W078-W102: 2011 Joint International Conference. Sophia Antipolis, France; 26-28 October. Eriřim adresi: <http://2011-cibw078-w102.cstb.fr/papers/Paper-20.pdf>
- Haagenrud, S.E., Björkhaug, L., Wix, J., Trinius, W., Huovila, P.** (2008). EU-Project STAND-INN Integration of Standarts for Sustainable Construction into Business Processes Using BIM/IFC, ECPPM 2008 eWork and eBusiness in Architecture, Engineering and Construction, ss. 487-495.
- Hirakawa, M. ve Ichikawa, T.** (1992). Advances in visual programming, Systems Integration, ICSI'92. ss. 538–543. Eriřim adresi: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=217309
- Holzer, D.** (2016). The BIM Manager's Handbook: Guidance for Professionals in Architecture, Engineering and Construction, United Kingdom, John Wiley and Sons.

- Hudson, M. ve Strathaus, T.** (2015). AS11047: Dynamo Hero 2 Using Revit Scripting to Optimize Real-World Projects, Autodesk University 2015. Erişim adresi: <http://au.autodesk.com/au-online/classes-on-demand/class-catalog/classes/year-2015/revit-for-architects/as11047#chapter=0>
- Humppi, H. ve Österlund, T.** (2011). Algorithm-Aided BIM, eCAADe 34 (2016) BIM Concepts, ss. 601–609.
- Kang, L. S. ve Paulson, B. C.** (2000). Information classification for civil engineering projects by Uniclass. Journal of Construction Engineering and Management, 158-167.
- Kensek, K. M.** (2014). Building information modeling, Abingdon; New York, Routledge.
- Kensek, K. M. ve Noble, D.** (2014) Building information modeling: BIM in current and future practice. Hoboken, New Jersey: Wiley.
- Keough, I.** (2015). Dynamo Now, Dynamo Next. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <http://www.formfollowsperformance.com/2015/05/a-case-webinar-dynamo-now-dynamo-next/>
- Koppula, S.** (2012). BIM Collaboration, Autodesk University. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2012/autodesk-vault-products/bim-collaboration#chapter=0>
- Kron, Z.** (2014). QveA about Dynamo, DynamoBIM. Erişim tarihi 28 Nisan 2018, Erişim adresi: <http://dynamobim.com/qa-about-dynamo/>
- Leitão, A. ve Santos, L.** (2011). Programming Languages for Generative design, 29th eCAADe Conference, Slovenia, ss. 549-557. Erişim adresi: http://cuminad.architecturez.net/system/files/pdf/ecaade2011_118.content.pdf
- Mandrill Primer.** (t.y.). What is this thing called LOD. Erişim tarihi 28 Nisan 2018, Erişim adresi: <https://legacy.gitbook.com/book/konradsobon/mandrill-primer/details>
- McPhee, A.** (2013). What is this thing called LOD. Erişim tarihi 28 Nisan 2018, Erişim adresi: <http://practicalbim.blogspot.com.tr/2013/03/what-is-this-thing-called-lod.html>
- MeLampy, M. F.** (2007). Specifications and MasterFormat 2004. PACE. SSPC.
- Menges, A. ve Ahlquist, S.** (2011) Computational design thinking. Wiley (AD Reader).
- Miettinen, R. ve Paavola, S.** (2014). Beyond the BIM utopia: Approaches to the development and implementation of building information modeling. Automation in Construction 43, ss. 84-91.

- Mihindu, S. ve Arayici, Y.** (2008). Digital construction through BIM systems will drive the reengineering of construction business practices, 12th International Conference Visualisation, IEEE Computer Society.
- Myers, P. S.** (1996). Knowledge management and organizational design. Boston, Mass.: Butterworth-Heinemann.
- National BIM Standard Version 1- Part 1: Overview, Principles, and Methodologies.** (2007). National Institute of Building Sciences, Erişim adresi:
https://buildinginformationmanagement.files.wordpress.com/2011/06/nbimsv1_p1.pdf
- Ofluoğlu, S.** (2008). Yapı Bilgi Modelleme: Gereksinim ve Birlikte Çalışabilirlik, Mimarist, 49, ss. 10-12.
- Ondejcik, V.** (2016). Dynamo Graphic Standards at White arkitekter AB. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <http://dynamobim.org/dynamo-graphic-standards-at-white-arkitekter-ab/>
- Penn State.** (2011). “BIM Project Execution Planning Guide and Templates - Version 2.1” BIM Project Execution Planning, CIC Research Group, Department of Architectural Engineering, The Pennsylvania State University
- RIBA ve ARUP.** (t.y). Designing with data: Shaping our future cities. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <https://www.architecture.com/-/media/gathercontent/designing-with-data/additional-documents/designingwithdatashapingourfuturecitiespdf.pdf>
- SCRA.** (2006). Step Application Handbook, ISO 10303, Version 3, N. Charleston
- Sgambelluri, M.** (2015). The Simply Complex Blog: WHAT IS DYNAMO? Erişim tarihi: 28 Nisan 2018, Erişim adresi: <http://therevitcomplex.blogspot.co.uk/2015/01/what-is-dynamo.html>
- Terzidis, K.** (2006). Algorithmic architecture. 1st ed.; Boston: Architectural Press.
- Tomasetti, T.** (2015). Introducing In-House BIM Workflow Apps, CORE studio. Erişim adresi: <http://core.thorntontomasetti.com/introducing-in-house-bim-workflow-apps/>
- TS-EN 12464.** (2013). Işık ve Aydınlatma- Çalışma Yerlerinin Aydınlatılması- Bölüm 1: Kapalı çalışma alanları, Türk Standartları Enstitüsü.
- Tuomi, I.** (1999). Data is more than knowledge: implications of the reversed knowledge hierarchy for knowledge management and organizational memory. Journal of Management Information Systems, 16, ss. 103-117.
- Vannini, A.** (2015). Acoustamo Released!. Erişim tarihi: 28 Nisan 2018, Erişim adresi: <https://andreaarch.wordpress.com/2015/02/15/acoustamo-released/>

- Weygant, R. S.** (2011). BIM Content Development: Standards, Strategies, and Best Practices, John Wiley ve Sons.
- Yoon, S., Park, N. ve Choi, J.** (2009). A BIM-Based Design Method for Energy-Efficient Building. International Conference on Networked Computing and Advanced Information Management, Seoul, ss. 376-381.
- Url-1** < <http://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Part-I-2017-11-07-1.pdf> >, erişim tarihi 28.04.2018.
- Url-2** < http://bimtalk.co.uk/bim_glossary:bim_dimensions >, erişim tarihi 28.04.2018.
- Url-3** < https://images.arcadis.com/media/5/F/2/%7B5F298123-F67E-4CAD-A01C-03A9C1C0CCFE%7DBIM_According_to_Arcadis_Asia_001.pdf >, erişim tarihi 28.04.2018.
- Url-4** < <https://damassets.autodesk.net/content/dam/autodesk/www/campaigns/test-drive-bim-q3/bds/tr/tr-fy16-const-champions-ebook-bim-solutions.pdf> >, erişim tarihi 28.04.2018.
- Url-5** < <https://info.bim360.autodesk.com/bim-360-glue> >, erişim tarihi 28.04.2018.
- Url-6** < <https://www.solibri.com/journal/kubus-launches-bcf-manager-%E2%80%A8for-revit-archicad/> >, erişim tarihi 28.04.2018.
- Url-7** < <https://dynamopackages.com/> >, erişim tarihi 28.04.2018.
- Url-8** < <http://www.msa-ipd.com/MacleamyCurve.pdf> >, erişim tarihi 28.04.2018.

ÖZGEÇMİŞ



Ad-Soyad : Ayşegül Özlem Bayraktar Sarı
Doğum Tarihi ve Yeri : 05.09.1991 / Trabzon
E-posta : aysegulozlembayraktar@hotmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2014, MSGSÜ, Mimarlık Fakültesi, Mimarlık Bölümü

MESLEKİ DENEYİM:

- İkikerebir Mimarlık - *Mimar* / Temmuz 2014 - Aralık 2014
- Dome+Partners - *Mimar* / Ekim 2015 - Ekim 2016
- GMW Mimarlık - *Mimar* / Ocak 2017 – Ocak 2018