

# The Integration of Computational Design with BIM: The Exploration of New Possibilities with “Dynamo”

Enes Kaan Karabay <sup>1</sup>

<sup>1</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

In this article, new opportunities emerged with the graphic programming language added on BIM will be examined. Open source Dynamo was chosen as the working environment. The new system created by Revit and Dynamo, was named as Computational BIM.

In the article, the stages of the study which includes Computational BIM in a design process will be explained. Autodesk has also added a plugin called “DynamoPlayer” to Revit. This plugin makes it easy to use written algorithms in a large group of designers and reduces the need for information; this opens the way for automation by the end user. As an advanced language created by Autodesk, “DesignScript” is integrated directly into Dynamo, and also Python scripts can be written directly into the Dynamo environment. With all its advantages, the Revit environment integrated with Dynamo offers a new paradigm shift as “Computational BIM”. Algorithms are used to execute commands on geometry and BIM database and to create logical and mathematical operations. With the help of these logical operations, the workforce required for recurring tasks can be reduced. Algorithm-supported design concept is generally used to define methods in computer environment. In the article, the concept of "computation" is used instead of "algorithm supported" due to the change in the knowledge-based system.

**Received:** 01.04.2020

**Accepted:** 02.04.2020

**Corresponding Author:**

[enes@kaankrby.com](mailto:enes@kaankrby.com)

Karabay, E. K. (2020). The Integration of Computational Design with BIM: The Exploration of New Possibilities with “Dynamo”, JCoDe: Journal of Computational Design, 1(2), 41-64.

**Keywords:** Building Information Modeling, Computational BIM, Computational Design.

# Hesaplmalı Tasarım Ve YBM Entegrasyonu: “Dynamo” İle Yeni Olanakların Araştırılması

Enes Kaan Karabay<sup>1</sup>

<sup>1</sup> İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Bu makalede, YBM üzerine eklenen grafik programlama dili ile ortaya çıkan yeni olanaklar incelenecektir. Açık kaynaklı Dynamo, çalışma ortamı olarak seçildi. Revit ve Dynamo'nun oluşturduğu yeni sistem Bilgisayimli YBM olarak adlandırılmaya başlandı.

Makalede, bir tasarım sürecindeki Hesaplmalı YBM içeren çalışmanın aşamaları açıklanacaktır. Autodesk, Revit'e “DynamoPlayer” adlı bir eklenti de eklemiştir. Bu eklenti, yazılı algoritmaların büyük bir tasarımcı grubunda kullanımını kolaylaştırır ve bilgi gereksinimini azaltır; bu da son kullanıcı tarafından yapılan otomasyonların yolunu açar. Autodesk tarafından oluşturulan gelişmiş bir dil olarak “DesignScript” doğrudan Dynamo ile bütünleştirilir ve ayrıca Python betikleri doğrudan Dynamo ortamına yazılabilir. Tüm avantajları ile Dynamo ile entegre Revit ortamı “Hesaplmalı YBM” olarak yeni bir paradigma değişimi sunar. Algoritmalar, geometri ve YBM veritabanı üzerinde komutları yürütmek, mantıksal ve matematiksel işlemler oluşturmak için kullanılmaktadır. Bu mantıksal işlemler sayesinde, yinelenen görevler için gereken iş gücü azaltılabilir. Bilgisayar ortamındaki yöntemleri tanımlamak için genellikle algoritma destekli tasarım kavramı kullanılır. Makalede, bilgi tabanlı sistemdeki değişim nedeniyle “algoritma destekli” yerine “hesaplama” kavramı kullanılmıştır.

**Teslim Tarihi:** 01.04.2020

**Kabul Tarihi:** 02.04.2020

**Sorumlu Yazar:**

[enes@kaankrby.com](mailto:enes@kaankrby.com)

Karabay, E., K. (2020). Hesaplmalı Tasarım ve Yapı Bilgi Modeli Entegrasyonu: „Dynamo“ ile Yeni Olanakların Araştırılması. JCoDe: Journal of Computational Design, 1(2), 41-64.

**Anahtar Kelimeler:** Yapı Bilgi Modelleme, Bilgisayimli YBM, Hesaplmalı Tasarım.

## 1. GİRİŞ

Teknoloji tarihinin günümüzdeki en büyük kırılma noktası bilgisayarın icadı ve yaygınlaşması olarak gösterilebilir. Bilgisayarın kişisel olarak edinilebilmesinin kolaylaşması ve internetin hayatımızın bir parçası haline gelmesi sonucunda bugün her alanda teknolojiden ve ürünlerinden yararlanmaktayız.

Bilgisayarın ve mobil aygıtların kullanımının artmasının yanı sıra internet devrimi ve “nesnelerin interneti (Internet of Things: IOT)” ile teknoloji artık hayatın tamamında yer etmeye başladı. İnternet devriminin bilgiye ulaşmayı kolaylaştırması sonucunda önceleri profesyonel olarak öğrenilen programlama yetkinliği ve elektronik araçların kullanımı gibi yeterlilikler, artan ilgi ile herkes tarafından ulaşılabilir olmuştur ve bu değişimleri de “Arduino” gibi elektronik araçları yapmayı kolaylaştıracak başka teknolojiler izlemiştir. Bugün artık kodlama ve mucitlik (maker) profesyonel bir uğraş olmaktan çıkmış ve yaş grubu olarak da çocuk denebilecek yaşlara inmiştir (ör: LittleBits2). Nesnelerin interneti ile nesnelerin arasında iletişim sağlanmıştır. Bu süreçlerin içerisinde ve sonucunda üretilen “veri” sadece bir yan ürün olmaktan çıkıp işlenmeye açık veri haline gelmiştir. Veri madenciliği sayesinde birçok sektörde istatistiksel veri, karar verme süreçlerinin bir parçası olmuştur.

Teknolojideki bu kırılmalar farklı şekillerde mimariye de yansımıştır. Bilgisayarın kullanımı önceleri iş gücünün kolaylaştırılması yönünde üretilmiş uygulamalarla karşımıza çıkmıştır. Gelişen araçlarla birlikte ise bilgisayarın tasarım, temsil ve üretim süreçlerinin içine girdiği hatta yönlendirdiği de görülmektedir. Bugün artık tasarım pratiğini de değiştiren bir kullanım olduğu aşikârdır.

Programlamaya olan ilgi artışının mimari platformda yansımaları da “Görsel Programlama Dillerinin (GPD)” mimari tasarım ortamlarında kullanılması olarak görülebilir. Mimarideki diğer teknolojik paradigma kırılmaları da katı modellemenin tasarımda kullanılması ile “Yapı Bilgi Modellemenin (YBM)” temellerinin atılmasıdır. Genel bir bakışla birbirlerinden ayrı ortamlarda gelişmeye devam eden bu kırılmaların, tek bir proje sürecinde kontrol edilmesi hem çeşitli zorluklar ortaya çıkartmakta hem de veri kayıplarına neden olabilmektedir. YBM ortamının tutarlı bir proje ortamı olarak kullanılmasının yanında, “Rhino- Grasshopper” birlikteliğindeki ön tasarım esnekliğini sunamıyor

oluşu, farklı ortamların paralel olarak aynı proje kurgusu içinde kullanımlarını zorunlu kılmaktadır. Bu makalenin amacı, GPD ile entegre olan YBM ortamının sağladığı yeni olanakları ortaya koymak ve incelemektir. Mevcut bir çevrede bir yapı tasarım modeli üzerinden, kapsam olarak çok genişletilebilecek bir Yapı Bilgi Modeli ile Görsel Programlama Dili olanaklarını bütünleştiren bir uygulama sürecinin geliştirilmesi hedeflenmiştir. Dynamo ile oluşturulan bu ortamın kullanımındaki en büyük motivasyon da farklı ortamlar yerine tek bir ortamda kurgulanan proje sürecini bize sunuyor oluşudur.

GPD ile bir betik oluşturma, grafik bağlantılar kullanılarak kodlamanın basitleştirilmesini sağlar. Bu basitlik ile, YBM ortamının kısıtlarının da dışına çıkabilen ve hayal gücü ile sınırlı yeni bir ortamın kullanımı ortaya çıkmaktadır. Yine Rhino – Grasshopper birlikteliği örneğinden yola çıkarsak, bu ortamda Grasshopper ortamın hesaplamalı tasarım alanındaki açığını kapatmaktadır. Rhino katı model yapan bir ortam olduğu için Grasshopper’da katı modelleme alanında açıkları kapatmaktadır. Revit – Dynamo birlikteliğinin hesaplamalı tasarım alanının dışına çıkabilmesinin nedeni ise, sadece modelleme alanında değil; aynı zamanda Revit’in YBM özelliklerine de eklenebilmesidir. Bu durumda da Dynamo ile Revit kullanımı Bilgisayimli YBM olarak adlandırılmaya başlanmıştır.

Makalede Bilgisayimli YBM aracı olarak Dynamo ele alınmış, örneklendirme ve çalışmalar da yine Dynamo üzerinden yapılmıştır. Basitçe ele alındığında, herhangi başka bir YBM ortamında GPD’nin bütün ortamı düzenleyebilecek şekilde entegre olmadığını görmekteyiz; var olan diğer araçlar da GPD araçları yerine daha çok belirli işlemler için özelleştirilmiş eklentiler şeklinde çalıştığından, Dynamo daha kapsamlı bir ortam olarak öne çıkmaktadır. Daha geniş bir perspektifte, GPD ve YBM entegrasyonu ucu açık ve sınırı YBM ortamının program kısıtlarına uzanan hatta bazı durumlarda bu sınırların da aşılmasını sağlayan, farklı ortamlarla veri kaybı olmaksızın ilişki kurmaya da yarayan bir ortam oluşturmayı sağlamıştır.

## **2. DYNAMO İLE TASARIM SÜRECİ**

Bu bölümde, bir tasarım sürecinde “Dynamo” aracı kullanılarak Bilgisayimli YBM yaklaşımı anlatılacaktır ve neden bilgisayarımı denildiği de ortaya koyulmaya çalışılacaktır. Yapılan örneklendirmelerle süreç

içerisinde Dynamo ve Revit birlikteliğinin neleri kapsadığı araştırılacaktır. Tasarıma konu olan modelin, Dynamo ortamında tasarlanması ve analize bağlı parametrelerle YBM ortamına aktarılması ile ilgili süreç açıklanacaktır.

Örneklenecek tasarım konusunun bina ihtiyaç programı:

- Sergi alanı olarak kullanılabilir kapalı bir alan;
- Fuaye alanı olarak açık mekân;
- Sergi alanları destek birimi olarak depolar;
- Kamusal toplanma alanı olarak pergola bölümü;
- Kafe alanı;
- Genel destek birimi olarak WC ve diğer alanlar olarak ele alınacaktır.

Tasarımın tüm aşamaları Dynamo görsel programlama dili kullanılarak bilgisayarlı bir ortamda gerçekleştirilecek ve süreçte sağladığı olanaklar tartışılacaktır.

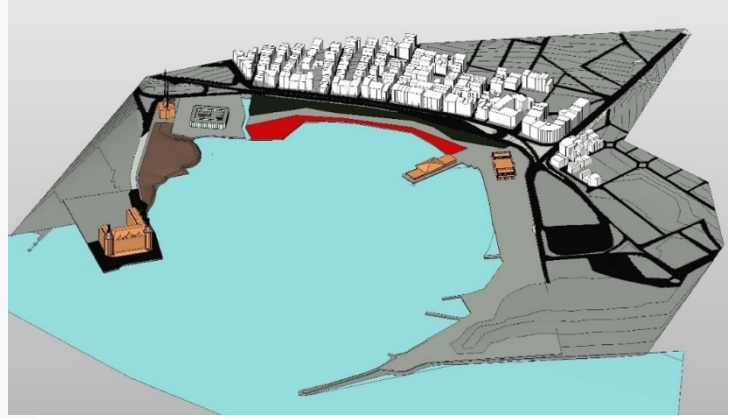
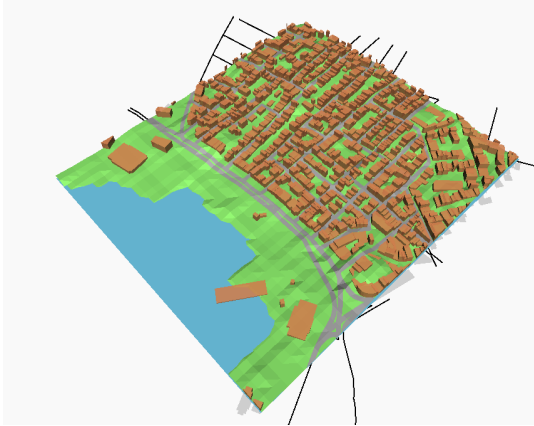
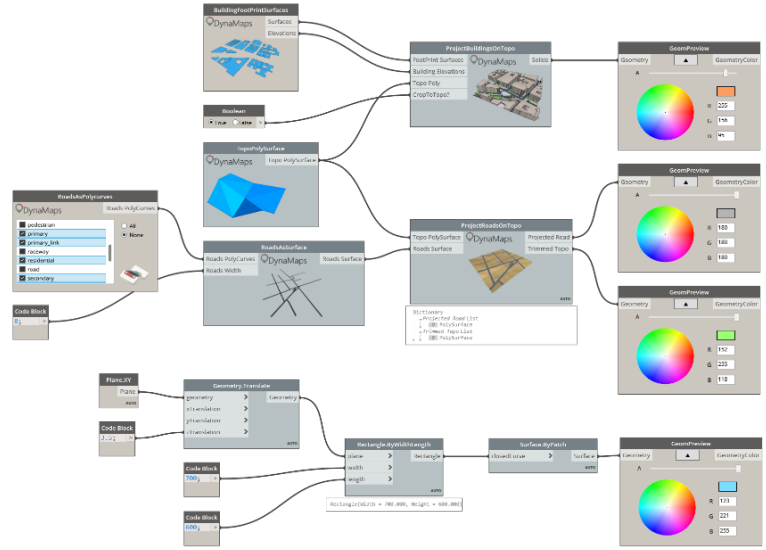
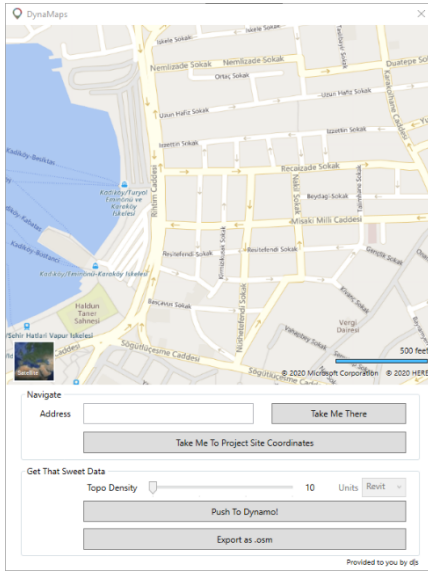
## **2.1 Ön Tasarım- Hesaplamalı Tasarım Aracı Olarak Dynamo**

Mevcut görsel programlama dilleri (GPD) arasında tasarım pratiği içerisinde en fazla destek bulanı ve kullanılanı şüphesiz Grasshopper eklentisi ile birlikte Rhino ortamıdır. Bu ortamın çok fazla destek bulma sebeplerinden biri de “açık kaynak” felsefesi ile geliştirilmesi ve paylaşılmasıdır. Açık kaynak kodlanan bir programın kendisi bir kamu ürünü haline geldiği için, yine geliştirilmesi de kamu tarafından olmaktadır. Bu durum aynı zamanda farklı disiplinlerde çalışan insanların bu tip programlara destek vermesini de kolaylaştırmaktadır. Dynamo da aynı şekilde açık kaynak olarak BuroHappold bünyesinde geliştirilmeye başlanmıştır; potansiyeli ve YBM bünyesindeki eksiklik olduğu için de daha sonrasında Autodesk bünyesinde yine açık kaynak olarak geliştirilmesine devam edilmiştir. GPD'nin bir tasarım aracı olarak kullanımı sayesinde tasarım ürününün girdileri üzerinden oluşturulan kontrol, tek defaya özgü bir form oluşturmanın yanı sıra, birçok analizi de tasarım girdisi olarak dahil etmeyi sağlayabilmektedir.

Bu bölümde, ön tasarım amacıyla gereken çevre verilerine, analizlere ve bunların girdi olarak nasıl değerlendirilebileceğine değinilecektir. Ön tasarım analizlerinde ilk adım, çevre modellerinin oluşturulmasıdır ve bu amaçla DynaMaps aracı kullanılmıştır. Bu araç sayesinde harita verileri Dynamo ortamına aktarılabilir. Aynı aracın bu verileri Revit ortamına aktarma özelliği de vardır. Bu sayede ortak veri

kütüphaneleri kullanılarak arazi altlığı oluşturulmuştur. DynaMaps aracı veri kütüphanesi olarak bir başka açık kaynak proje olan "Openstreetmap" 'i kullanmaktadır. **Şekil 1'**deki algoritma Dynamaps ile Dynamo ara yüzünde gerekli aktarımlar yapılır. Dynamo üzerinden aktarılan verilerle birlikte farklı haritalardan toplanan veriler birleştirilerek, Kadıköy meydan ve çevresi modeli oluşturulmuştur (**Şekil 2**).

**Şekil 1:** Dynamaps Aktarım Şeması ve Harita Seçim arayüzü.

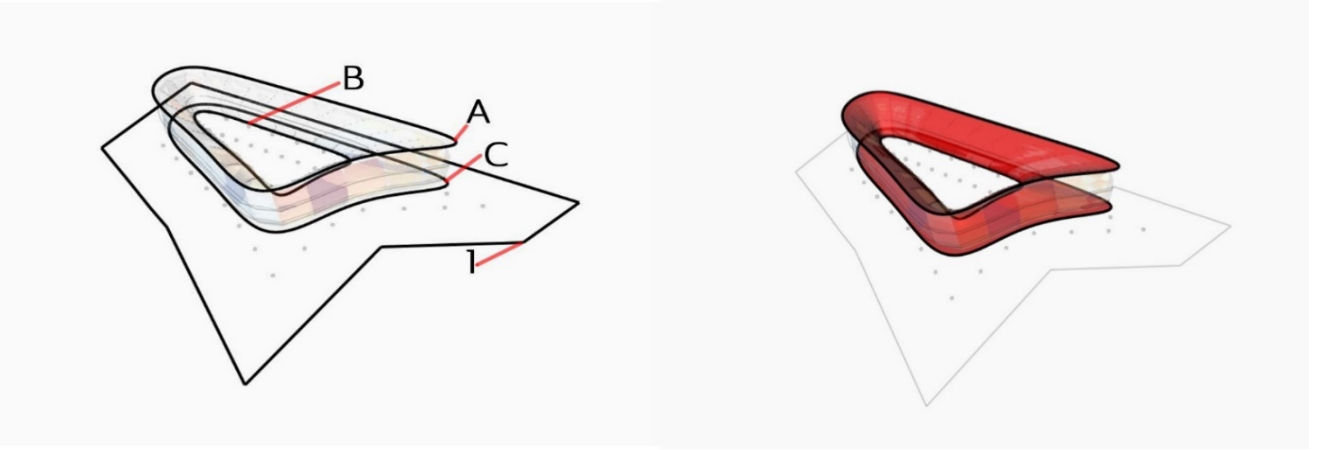
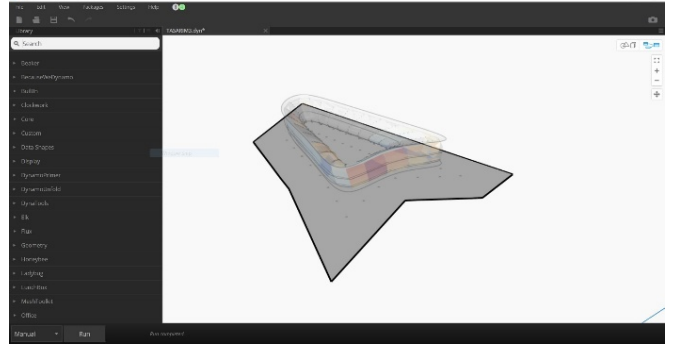
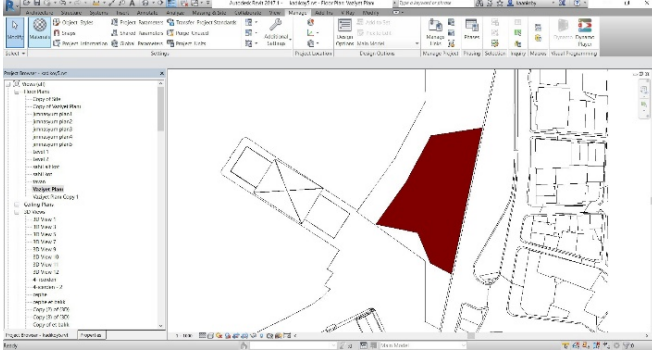


Tasarıma başlangıç aşamasında tasarlanacak yapı meydana belirli bir alan ile sınırlandırılmıştır. Bu alan "Select Face" nodu kullanılarak Dynamo ortamına aktarılmıştır. **Şekil 3'**de Revit ortamındaki ve Dynamo ortamına aktarılmış olan tasarım alanı görülmektedir Alanı oluşturan sınırlar Dynamo ortamında ayıklanarak "NurbCurve" haline getirilmiştir.

**Şekil 2:** DynaMaps ile oluşturulan Dynamo vaziyet planı modeli ve Revit ortamında birleştirilmiş olan model

**Şekil 3:** Revit ekranındaki belirlenmiş olan yapı analı sınırı ve Dynamo ortamına taşınmış olan tasarım alanı sınırı.

Oluşturulan Nurb Curve üç farklı kota yansıtılmış ve bunların birleştirilmesinden kütleye altık oluşturacak yüzey oluşturulmuştur. **Şekil 4**'de oluşturulan altık yüzeyinin geometrik temsili Dynamo ekranında renklendirilmiştir



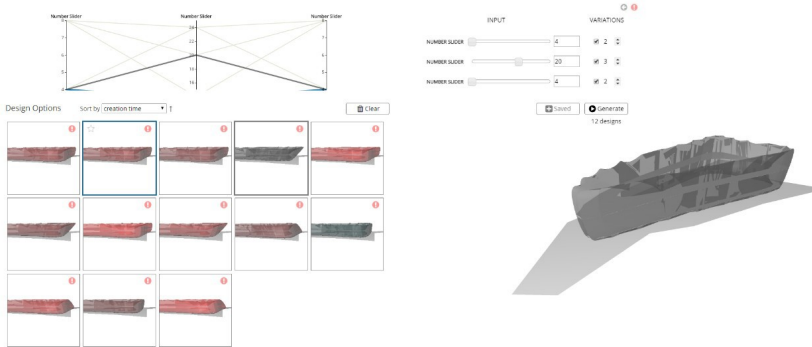
**Şekil 4:** Algoritma ile oluşturulan altık yüzeyi ve onu oluşturan Nurb Curvler.

Altık yüzey üzerinden de yüzeye dik olarak oluşturulan profiller birleştirilerek oluşturulan kütle görülmektedir (**Şekil 4**). Bu noktaya gelene kadar oluşturulan tüm geometrik elemanlar değişkenler yardımıyla oluşturulduğu için her parametre değişimi, sonucu değiştirmektedir. Aynı şekilde vaziyet planında belirlenmiş olan sınırlar değıştikçe de son form değışiklik gösterecektir.



Project Refinery üzerinden deęişkenler ile çeşitlendirme denemesi yapılmıştır. **Şekil 6**'da kütle rengi, toplam kat alanına göre deęişkenlik göstermektedir. Verinin bu şekilde görselleştirilmesi ile sonuçlar arasında ayırım yapılarak en uygun olan geometrik sonucun seçilmesi amaçlanmaktadır. Bu basit çeşitlendirmenin yanı sıra Refinery daha kapsamlı Genetik çözümler de yapılmaktadır.

**Şekil 5:** Liste içerisinde oluşturulan ve profiller ve birleştirilen profiller.



**Şekil 6:** Project Refinery Çeşitlendirmeleri.

## 2.2 Şematik Tasarım – Revit Ortamına Yapı Elemanlarının Eklenmesi

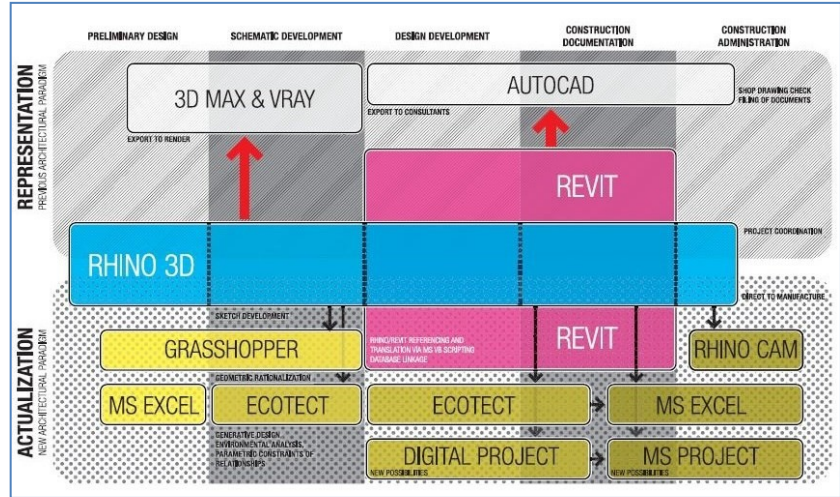
Ön tasarım bölümünde kullandığımız araçların benzerlerini başka tasarım ortamlarında da görmekteyiz. Ön tasarımdan yapı elemanlarından oluşan bir YBM'ye geçişte Dynamo ortamındaki geometrinin doğrudan katı model olarak aktarımının yapılabileceği gibi aynı zamanda Revit doğal elemanı araçları da kullanılarak model oluşturulabilir. İlk ciddi farklılığı sağlayan da Dynamo'nun bu etki alanındaki genişliktir. Grasshopper ortamında şu an için Dynamo ortamının çok önünde hesaplamalı tasarım modelleri oluşturulabilir,



fakat oluşturulan model katı bir modelleme ve üç boyutlu yapı temsilinden öteye geçemeyecektir. Bu ortamdan yapılan bir aktarım sırasında da birçok veri aktarım problemi yaşanabileceği gibi tasarım modelinin aktarımı sırasında da büyük bir iş gücü kaybı yaşanacaktır. Bazı durumlarda bütün modelin tekrar YBM içerisine işlenmesi bile gerekmektedir. Çoğu durumda ise tasarım sürecinin önceden belirlenmiş bir bitiş zamanı olmadığı için şematik tasarım ve daha sonraki aşamalar için Revit ortamına paylaşılan model, Rhino ortamında da tasarım devam ettiği için sürecin gerisinde kalmaktadır. Bu durum süreç içerisinde uyumsuzluklar ortaya çıkartmaktadır.

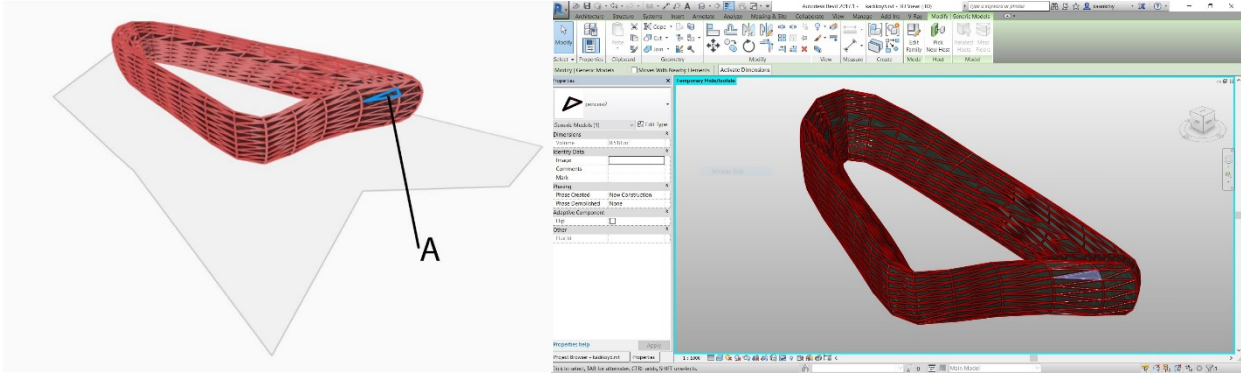
Geleneksel bir proje aktarım stratejisinde (Şekil 7) bu noktada oluşan kütle ya da daha nitelikli model, farklı dosya formatlarında uyumluluklar aranarak YBM ortamına aktarılmaya çalışılır. Çoğunlukla YBM ortamı ile tasarım modeli ortamı birbirlerine paralel olarak iki farklı koldan ilerlemeye çalışır ve bu iki ortamın koordine olması beklenir. Gerçekte ise bu koordinasyon çoğunlukla birbirinden kopar veya en iyimser haliyle bile anlık bir aktarım olmadığından sürekli koordinasyona muhtaçtır.

**Şekil 7:** GRO Mimarlık tarafından uygulanan Hesaplamalı Tasarım ve Yapı Bilgi Modelleme Çalışma Şeması (Garber, 2014).



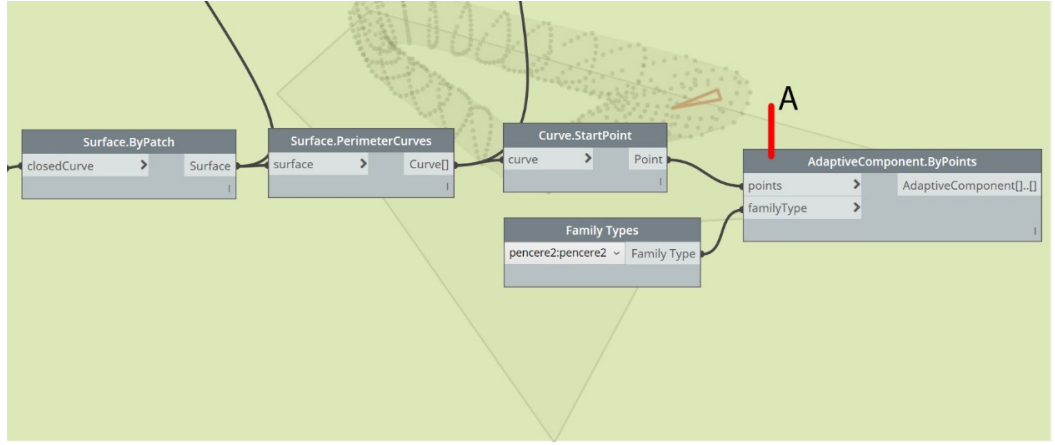
Başka bir ortamda hazırlanmış olan katı bir modelin içe aktarılıyor olmasının aksine, Dynamo ile Revit ortamında nesnel oluşturmanın en büyük avantajı, değişkenler girilerek veri tabanı oluşturulmasıdır. Revit ve Dynamo ortamının oluşturduğu birlikteliği, Grasshopper ve Rhino ortamından ayıran en büyük fark da bu noktada ortaya çıkmaktadır.

Bu ortamı Bilgisayımli YBM yapan yönü ise, Dynamo'nun Revit YBM veri tabanına etki edebilme kısmıdır. Bu birliktelik, yalnızca iki sürecin paralel çalışması şeklinde değildir; ortam kısıtlarının bir diğeri tarafından kapatılabilmesini de sağlamaktadır. Tasarım modeli üzerinden bunu ele alırsak, Dynamo ortamında oluşturulan tasarım girdileri Revit ortamına aktarılarak modelin oluşturulması sağlanmıştır. kullanılarak Revit veri tabanında elemanların oluşturulması sağlanmıştır. **Şekil 8**'de Revit ortamında eklenmiş olan cephe panelleri görülmektedir.



Bu sayede tasarım katı bir model yerine YBM olarak oluşturulmuştur ve bu geçiş bir sürecin devamı olduğu için herhangi bir veri kaybı da olmayacaktır. Tasarımın ilk aşamasında oluşturulan kütleler yine sürecin devamında kullanılmaya devam edilmiştir. Oluşturulan kütle cephe panelleri için bir altlık yüzey olarak kullanılmıştır. Cephe panellerinin yerleştirileceği yüzey ilk olarak bölümlenmiş ve panel aralıkları Dynamo ortamında belirlenmiştir. Paneller için gerekli altlık yüzeyleri oluşturulmuştur ve **Şekil 9**'da örnek bir yüzey "A" ile işaretlenmiştir. Paneller Revit ortamında oluşturulurken "Adaptive component" aracı kullanılmıştır. Bu araç, hazırlanan panellerin köşe noktalarının doğru sırayla verilmesi durumunda Revit ortamında veri tabanına eklenmektedir. Dynamo ortamında da bu aracı kontrol eden bir "node" bulunmaktadır. **Şekil 9**'da "A" ile işaretlenmiş olan bu araç, gruplanmış halde köşe noktalarını ve Revit "family" girdi olarak objelerin yerleştirilmesini sağlamaktadır. Panellerin köşe noktaları aynı sırayla gruplanmalıdır. Bu işlem için Revit 'panel family' dosyası hazırlanmış ve Dynamo ortamında da yüzeyler hazırlanıp köşe noktaları yakalanmıştır (**Şekil 8**). Bu noktalar kullanılarak Revit veri tabanında elemanların oluşturulması sağlanmıştır.

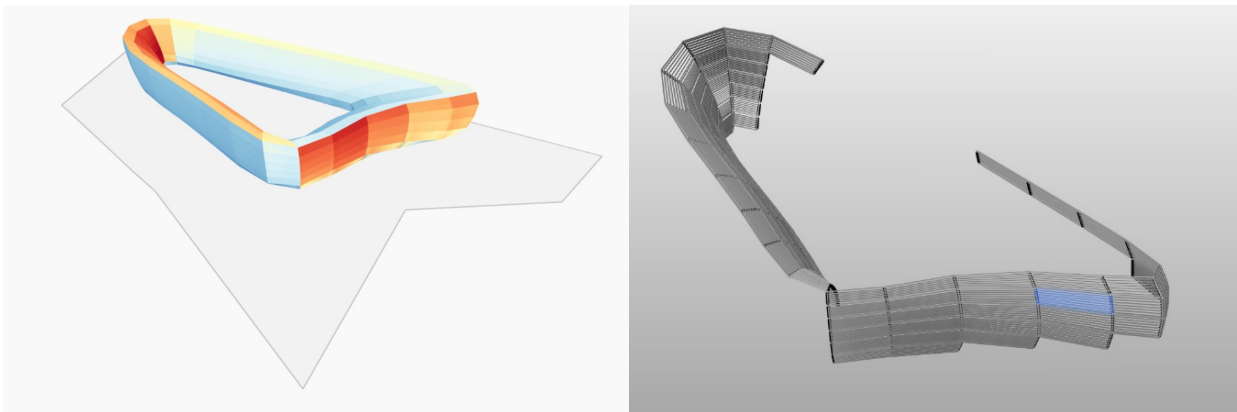
**Şekil 8:** Panel altlığı yüzeyler ve çerçeveleri.



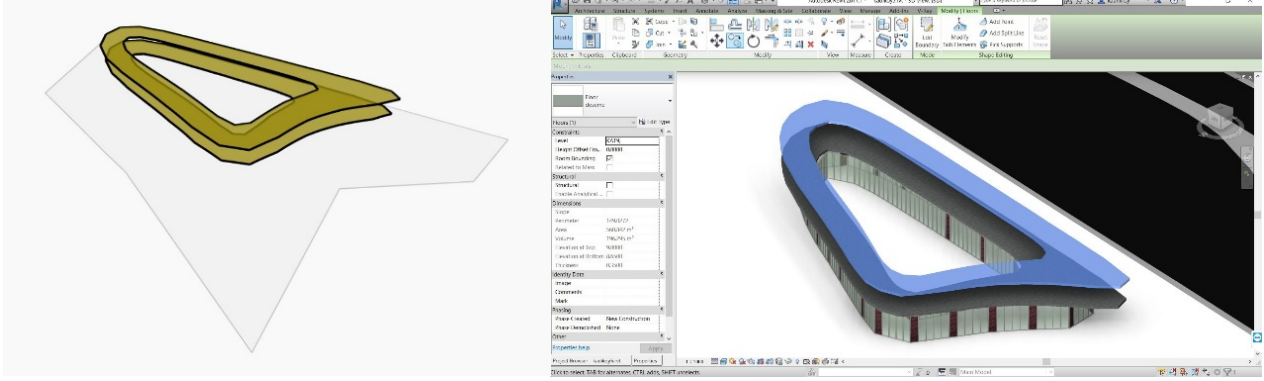
**Şekil 9:** Dynamo ortamında Adaptive Component aracı.

**Şekil 8'**de Revit penceresinde seçili olan (mavi renkli) Revit elemanın ismi ve özellikleri sol pencerede görülmektedir. Bu objeler Revit veri tabanına eklendiği için sorgulama ve listelemelerde de görüleceklerdir. Tasarımın bu aşamasında projeye ikincil bir dış cephe katmanı güneş kırıcıları olarak yerleştirilmiştir. Panelizasyon işlemi sırasında aynı yöntem izlenmiş (**Şekil 10**) fakat farklı olarak yerleştirme işlemi, güneş kırıcılar için oluşturulan yüzey altlığında güneşlenme analiz (**Şekil 10**) sonuçlarına göre yapılmıştır. Bu analize bir yıllık aralık, lokasyon ve yüzeyler girdi olarak verilmiştir. Sonuçları renk skalası ile örtüştürülmüş ve maviden kırmızıya giden bir yüzey analizi elde edilmiştir. Bu analiz sonucu, Dynamo ortamında **Şekil 10'**daki gibi görülmektedir. Bu renklendirmeye örtüştürülen bir filtreleme ile de belirli bir değerin üstündeki paneller seçilmiş ve Revit ortamında bu seçilmiş olan panellerin yerleştirilmesi sağlanmıştır (**Şekil 10 – sağdaki şekil**).

**Şekil 10:** Dynamo ortamında Solar Analiz sonunda renklendirilmiş yüzeyler ve bu yüzeylere Revit ortamında denk gelen güneş kırıcılar.



Güneşlenme analizinin sonucunda yerleştirilen paneller Revit ortamında Şekil 10 – sağdaki şekil gibi görülmektedir. Yine bu panellerin her biri bağımsız objelerdir, tüm sorgulamalarda görülebilirler. Sonraki aşamada, Dynamo 'da oluşturulan kütlelerin Revit kat izleriyle kesişimi yakalanarak (Şekil 11), kat döşemelerinin de yine Dynamo betiğinin devamında eklenmesi sağlanmıştır. Bu kesiştirme sonucunda yakalanan izlerin bulunduğu kotlarda Revit katları ve izlerle de döşemeler veri tabanı içerisine yazılmıştır



Son olarak zemin kattaki dış duvarlar da aynı izler referans alınarak Dynamo ortamında oluşturulmuştur. Giydirme cephe şeklinde oluşturulan dış duvarların belli bir oran içerisinde kalan sayı kadarı, rastlantısal olarak seçilmiş ve cam panelleri taş malzemeli duvar panellerine çevrilmiştir. Şekil 11'de Revit ortamında eklenmiş olan döşemeleri ve duvar panelleri görülmektedir. Yine seçili olan döşeme özelliklerine bakıldığında Revit içerisinde modele elle girilen bir elemandan farksız olduğu görülmektedir.

Şekil 11: Dynamo ortamında kat izleri ve Revit ortamında aynı izlerin aktarılmış hali.

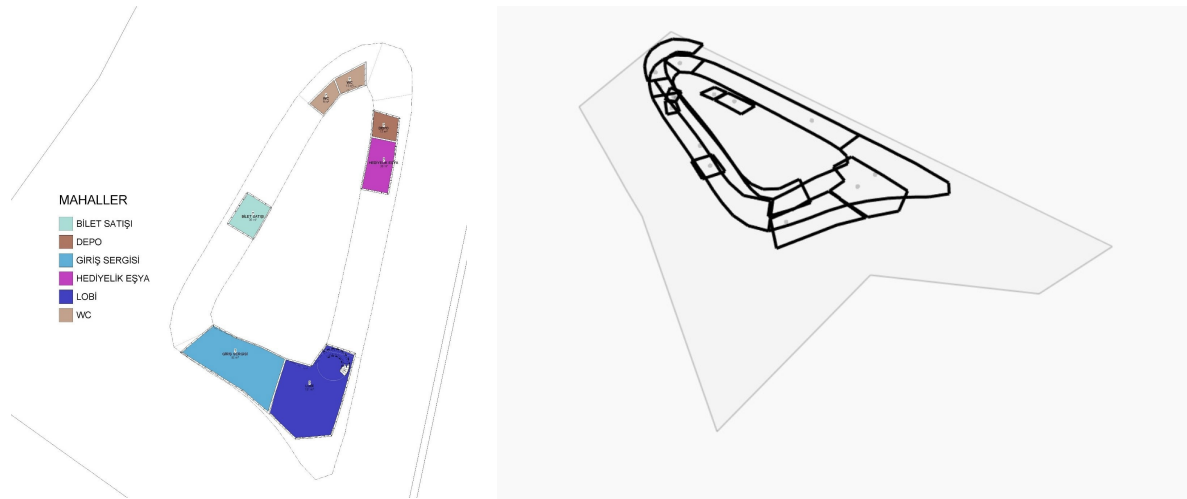
## 2.3 Tasarım Geliştirme

### 2.3.1 Oluşturulan mahallere göre döşeme izleri

Revit ortamının mahal aracı, duvar sınırlarını tanıyarak arasında kalan alanı tanımlar. Bu mahallere duvar, tavan malzeme bilgileri girilebilmektedir ve mahaller ön tanımlı olarak alan ve hacim bilgilerini veri tabanından okur.

Şekil 11'de görülen duvarlar Dynamo betiği ile oluşturulmuştu. Zemin katta boşluklar oluşturmak adına bazı sınır duvarları kaldırılmıştır. Üst katlarda da galeri boşluklarının açılması gibi fonksiyona bağlı yorumlar

yapılmıştır. Yapılan düzenlemeler sonrasında mahaller elle eklenmiştir. Tasarım geliştirme sırasında eklenecek olan ince yapı döşemeleri için Dynamo ile oluşturulan betik kullanılmıştır. Betik yardımı ile mahaller Revit ortamından okunarak sınırları bulunmuş (Şekil 12) ve mahal sınırlarına uygun olarak döşeme izlerinin standart bir döşeme ile oluşturulması sağlanmıştır. Betik içerisinde mahallerin ait oldukları katlar veri tabanından alınmış ve döşemelerin bu kat isimlerine göre doğru kata döşeme üst kotundan 10 cm üste eklenmeleri sağlanmıştır ve eklenen 10 cm kalınlığındaki bitiş döşemeleri olmaları gerektiği kotlara yansıtılmıştır. Mahallerin sadece kat izleri değil, lokasyonları da veri tabanından okunabilmektedir. Betik yardımıyla döşeme izleri eklendiğinde, bitiş döşemelerinin eklenmesi kullanıcıya bağlı olan bir eylem olmak yerine otomasyona dönüşmektedir. Mahaller, duvar ve kolon birleşimlerini otomatik olarak tanıyarak konumlandığı için de tam doğru şekilde modellenmiş olmaktadır.



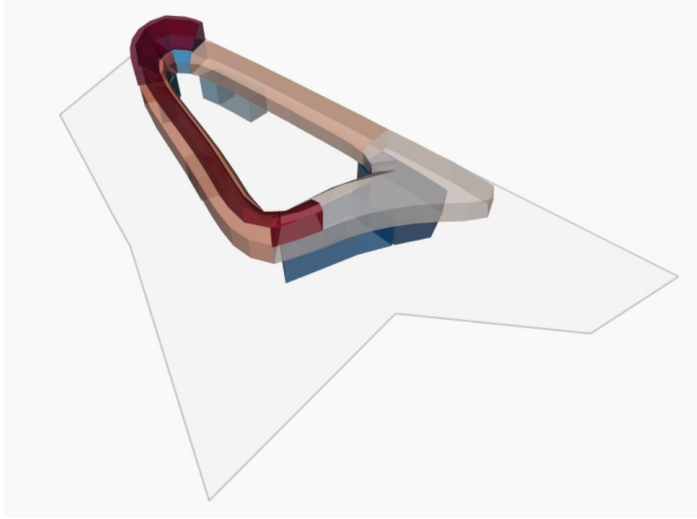
**Şekil 12:** Revit ortamında oluşturulan mahaller ve Dynamo ortamına aktarılmış mahal sınırları.

### 2.3.2 Üç boyutlu mahal renklendirme

Veri görselleştirme çalışmaları, proje geliştirme sırasında önemli yer tutan alanlardan bir tanesidir ve geleneksel hale geldiği kabul edilebilecek şekilde grafik düzenleme araçları ile yapılmaktadır. Proje içerisinden model görüntüleri veya kesitler, grafik düzenleme ortamına paylaşılarak, görselleştirmeler hazırlanmaktadır. Dynamo ortamı ile işlemlerin otomasyonunu sağlamak mümkündür. Mahalleri, Revit ortamında plan düzleminde iki boyutlu temsillerle isimlere veya departmanlara göre renklendirmek mümkündür. Fakat veri tabanında üç boyutlu objeler olarak bulunan mahallerin yine üç boyutlu olarak

renklendirilmesi Revit ortamında mümkün olmamaktadır. Dynamo ortamında bu objelerin geometrileri sorgulanarak, renklendirme çalışmasının temel alınacağı değişken olarak mahal isimleri, mahal departmanları, blok isimleri gibi farklı girdiler seçilir ve yapılan gruplandırma ile **Şekil 13**'deki gibi bir çalışma oluşturulabilir.

Algoritma geliştirilmesi sırasında oluşturulan bazı 'node'lar paket yöneticisi içerisine "ParametricAlly" paketi adı altında yüklenmiştir ve kullanıma açılmıştır.



**Şekil 13:** Dynamo ortamındaki mahal renklendirme çalışması.

## 2.4 Detaylandırma

### 2.4.1 Mahallerin numaralandırılması

Mahal veya otopark numaraları gibi değerler projelerde her zaman belirli bir sıra içerisinde düzenlenmektedir. Bu tip düzenlemeler de elemanlar üzerinde tek tek yapılmaktadır. Dynamo kullanarak YBM veri tabanındaki numaraların belirli bir otomasyon içerisinde düzenlenmesi sağlanabilir. Bu otomasyona referans olacak bir yol çizgisi belirlenerek bu çizgi üzerinden kesişen bütün elemanların kesişme sırasına göre numaralandırılması sağlanarak, kullanıcıların kendi girmeleri gereken parametreler yine bir otomasyon sayesinde hatasız bir şekilde projeye işlenmesi sağlanmaktadır.

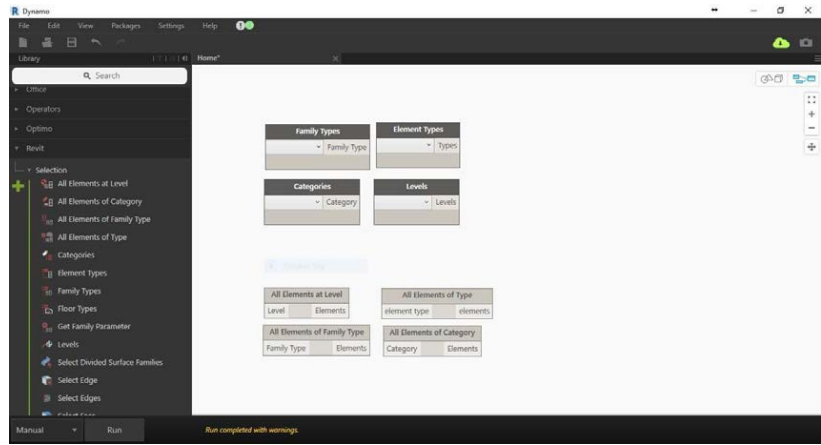
Hazırlanan betiğe girdi olarak düzenlemenin yapılacağı kat ve düzenlemeye referans olacak "curve" verilir. Mahallerin veri tabanından okunan lokasyon noktaları referans üzerine yansıtılır,

yansıtıldığı referans çizgisine göre orantısal konumları mahallerin sırasını belirtmektedir. Sıralamaya göre yeni numaralandırma formülü yapılmış ve numaralandırma işlemi sağlanmıştır. Bu basit örnek aslında yapılabilecek farklı otomasyonlar için fikir vermektedir.

#### 2.4.2 Dynamo ile veri tabanı kontrolleri ve çapraz parametreler

Revit elemanlarının hepsi aslında bir programlama dilinin parçaları gibi çalışmaktadır. Nasıl bir program kodunun çalışması için girdiler gerekliyse ve kod bu girdileri derleyip bize bir sonuç çıkartıyorsa, Revit araçları da girdileri alıp bize grafiksel bir sonuç verir ama asıl arka planda yaptığı şey programın bunu bir veri tabanı olarak saklamasıdır. Duvar oluşturmak için, bağlı olduğu alt ve üst kotlar, bu kotlardan ne kadar üst veya alta uzayacağı gibi parametreler girdi olarak alınır ve duvar katı bir eleman olarak üretilir. Duvarın bir sonuç ürünü olmasının yanı sıra bu duvarın ürettiği alan, uzunluk, yükseklik gibi diğer yan veri ürünleri de aynı veri tabanı içerisinde tutulur. Dynamo ile bahsedilen tüm parametreler, veri tabanı içerisinde tüm elemanlar için sorgulanabilir. Aynı zamanda Revit elemanlarını seçip Dynamo tarafından tanınır hale getiren seçim araçları da mevcuttur. **Şekil 14'**de Dynamo içerisinde bulunan "Revit eleman Seçim" araçları görülmektedir. Bu araçlar ile veri tabanına yazılmış olan bütün elemanların değişkenleri okunabilmektedir. Bu değişkenler de başka elemanlara veri olarak girilebilmektedir. Okunabilen parametreler program ara yüzünde görüldenden daha fazlasıdır.

**Şekil 14:** Dynamo ara yüzündeki Revit Eleman Seçimlerinden bazıları.



Daha somut bir örneklendirme gerekirse, projede bütün duvarların kat parametrelerine bağlı olması beklenmekte ise, aksi durumda olan duvarların üçüncü boyut ortamında renklendirilmesi sağlanabilir; hatta

hatalı olan bütün duvarların kat parametresine bağlanması sağlanabilir. Bu tip sorgulamalar ve kontroller Dynamo benzeri bir yapı kullanılmadığı takdirde otomasyondan uzak bir şekilde kullanıcı kontrolünde yapılmaktadır.

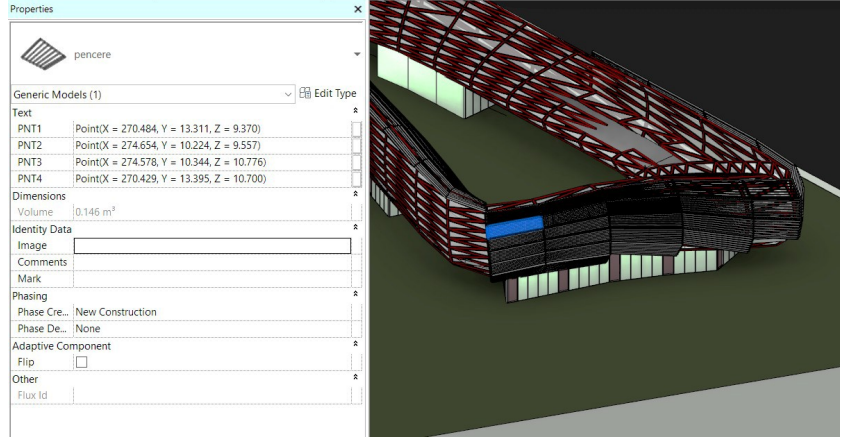
Veri tabanı üzerinden çapraz olarak değişkenlerin kullanılması için oluşturulan betikler ise, özellikle mantıksal olarak kurgulanabilen sıralı işlemlerin otomasyon haline getirilebilmesini sağladığı için, zaman açısından büyük avantajlar sağlar. Özellikle nitelikli iş gücünün aktif kullanılması açısından da bu tip işlemleri otomasyon haline getirmek önemli bir avantajdır.

Değişkenlere girdi eklemek iç için en iyi uygulamalardan biri mahal kodlarının farklı elemanlara değişken olarak eklenebilmesidir. Proje yönetimi sırasınca mahale bağlı olan kapıların kodlarının proje ortamına girilmesi sırasında, bağlı olduğu mahal kodlarının da parametre olarak eklenmesi istenmektedir. Böyle bir durumda bu kodların hem her mahal kodu değişikliğinde elle kontrol edilmesi zordur, hem de çok bloklu ve katlı projelerde girilmesi gereken parametre sayısı çok fazla artmaktadır. Örneğin Dynamo ile mahal kodlarının okunup mahale bağlı olan kapılara sırasıyla yazılması ve numaralandırılması sağlanabilir. Bu tip mantıksal fonksiyonlar kurulabilen işlemler, bir kişi için belki birkaç günlük iş gücü kaybı demek iken, Dynamo ile hazırlanan betik ile dakikalar içerisinde bitirilebilmektedir. Tasarım modelinde benzer bir problem olarak cephe panellerinin köşe koordinatlarının obje içine yazılması ele alınmıştır.

Hazırlanan panel objesinin özelliklerine eklenmiş olan “Pnt” değişkenlerine sırasıyla köşe noktalarının koordinatları eklenecektir. Köşe koordinatlarının normal araçlar ile Revit içerisinde yazılması mümkün değildir. Bu çözüm aynı zamanda program kısıtlarının önüne geçmek açısından da değerlendirilebilir. **Şekil 15'**de oluşturulan betik ile köşe noktaları koordinatları yakalanarak Revit veri tabanına girilmektedir.

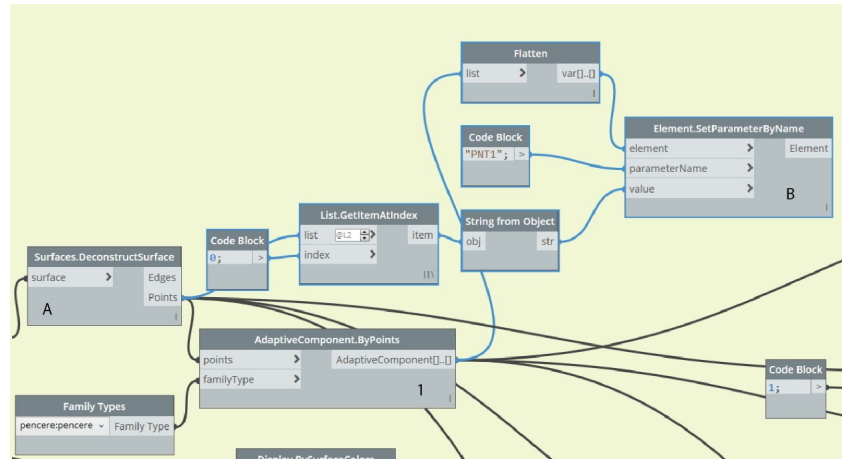


Şekil 15: Revit ortamındaki parametreler.



Şekil 16'da seçili olan güneş kırıcı panelinin 'family' özellikleri soldaki özellikler penceresinde görülmektedir. Betik yardımı ile verileri toplanan köşe noktaları "PNT1-2-3-4" altında girdiler halinde görülmektedir. Burada oluşturulan paneller, YBM veri tabanına eklenmiş objeler olduğu için ilgili tüm veriler Revit ortamındaki sorgulamalarda ve listelemelerde görünürler (Şekil 17). Dynamo ile Revit sorgulama araçları kullanılarak her türlü değişkenin okunması sağlanabilmektedir. Dynamo ile Revit sorgulama araçları kullanılarak her türlü değişkenin okunması sağlanabilmektedir.

Şekil 16: Panel ilk köşe noktasını okuyan betik.



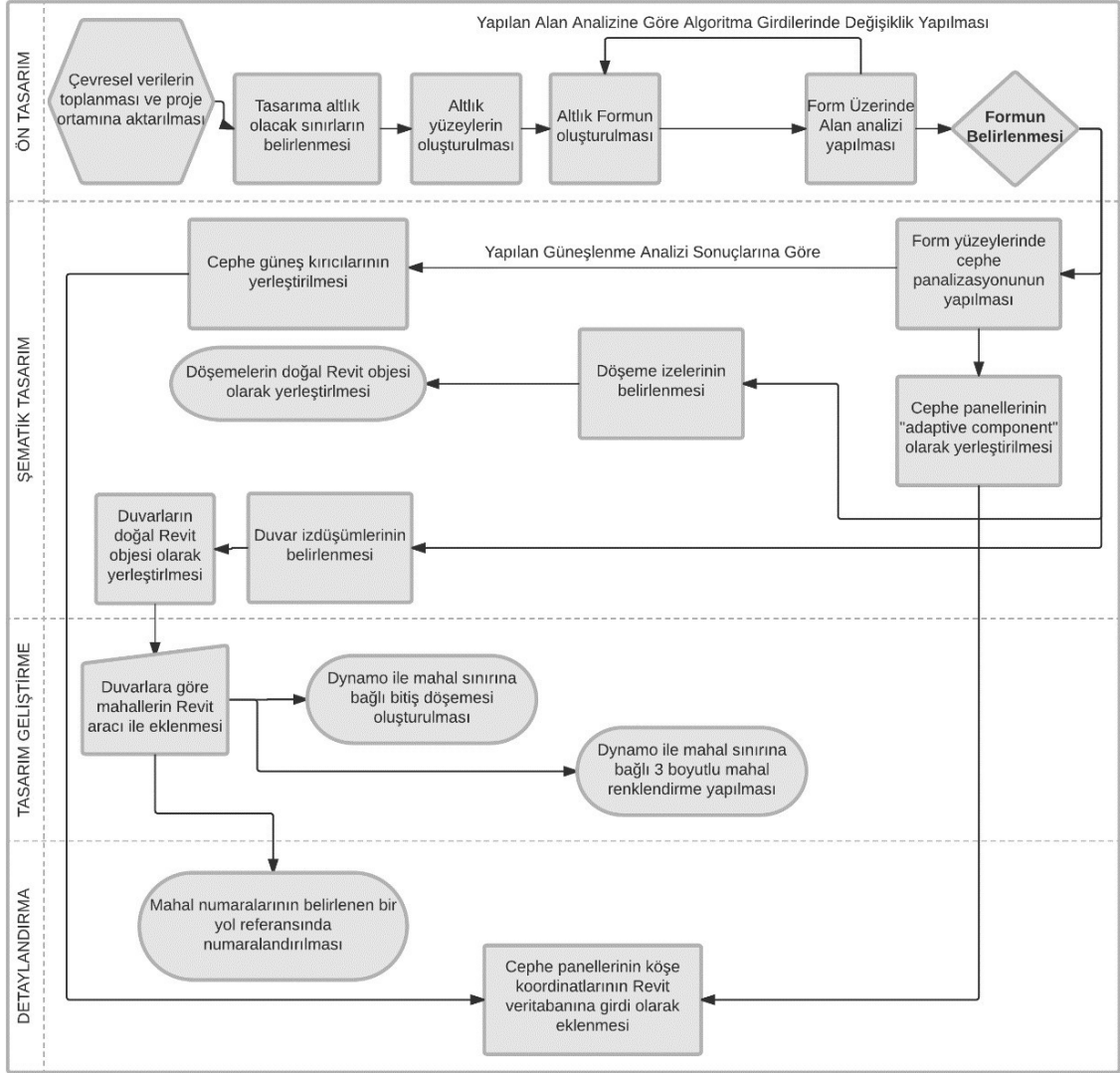
<Multi-Category Schedule>				
A	B	C	D	E
Family	PNT1	PNT2	PNT3	PNT4
pencere	Point(X = 296.965, Y = 7.663, Z = 10.519)	Point(X = 298.541, Y = 15.138, Z = 10.616)	Point(X = 298.334, Y = 15.403, Z = 12.070)	Point(X = 296.757, Y = 8.106, Z = 12.135)
pencere	Point(X = 298.541, Y = 15.138, Z = 10.616)	Point(X = 300.115, Y = 22.625, Z = 10.704)	Point(X = 299.912, Y = 22.748, Z = 12.002)	Point(X = 298.334, Y = 15.403, Z = 12.070)
pencere	Point(X = 304.442, Y = 44.744, Z = 11.968)	Point(X = 304.998, Y = 51.163, Z = 12.003)	Point(X = 303.144, Y = 49.951, Z = 11.473)	Point(X = 302.566, Y = 44.232, Z = 11.437)
pencere	Point(X = 304.998, Y = 51.163, Z = 12.003)	Point(X = 303.332, Y = 55.312, Z = 12.014)	Point(X = 301.769, Y = 53.547, Z = 11.436)	Point(X = 303.144, Y = 49.951, Z = 11.473)
pencere	Point(X = 303.144, Y = 49.951, Z = 11.473)	Point(X = 301.769, Y = 53.547, Z = 11.495)	Point(X = 300.968, Y = 52.671, Z = 10.833)	Point(X = 302.204, Y = 49.360, Z = 10.788)
pencere	Point(X = 302.204, Y = 49.360, Z = 10.788)	Point(X = 300.968, Y = 52.671, Z = 10.833)	Point(X = 300.180, Y = 51.813, Z = 10.133)	Point(X = 301.284, Y = 48.786, Z = 10.063)
pencere	Point(X = 301.284, Y = 48.786, Z = 10.063)	Point(X = 300.180, Y = 51.813, Z = 10.133)	Point(X = 299.411, Y = 50.981, Z = 9.381)	Point(X = 300.396, Y = 48.235, Z = 9.282)
pencere	Point(X = 300.396, Y = 48.235, Z = 9.282)	Point(X = 299.411, Y = 50.981, Z = 9.381)	Point(X = 298.677, Y = 50.194, Z = 8.551)	Point(X = 299.563, Y = 47.727, Z = 8.417)
pencere	Point(X = 299.563, Y = 47.727, Z = 8.417)	Point(X = 298.677, Y = 50.194, Z = 8.551)	Point(X = 298.020, Y = 49.506, Z = 7.583)	Point(X = 298.857, Y = 47.312, Z = 7.406)
pencere	Point(X = 298.857, Y = 47.312, Z = 7.406)	Point(X = 298.020, Y = 49.506, Z = 7.583)	Point(X = 297.644, Y = 49.161, Z = 6.352)	Point(X = 298.599, Y = 47.211, Z = 6.165)
pencere	Point(X = 303.332, Y = 55.312, Z = 12.014)	Point(X = 299.002, Y = 56.139, Z = 12.010)	Point(X = 298.089, Y = 53.976, Z = 11.496)	Point(X = 301.769, Y = 53.547, Z = 11.495)
pencere	Point(X = 301.769, Y = 53.547, Z = 11.495)	Point(X = 298.089, Y = 53.976, Z = 11.496)	Point(X = 297.610, Y = 52.891, Z = 10.836)	Point(X = 300.968, Y = 52.671, Z = 10.833)
pencere	Point(X = 300.968, Y = 52.671, Z = 10.833)	Point(X = 297.610, Y = 52.891, Z = 10.836)	Point(X = 297.136, Y = 51.827, Z = 10.139)	Point(X = 300.180, Y = 51.813, Z = 10.133)
pencere	Point(X = 300.180, Y = 51.813, Z = 10.133)	Point(X = 297.136, Y = 51.827, Z = 10.139)	Point(X = 296.673, Y = 50.794, Z = 9.390)	Point(X = 299.411, Y = 50.981, Z = 9.381)
pencere	Point(X = 299.411, Y = 50.981, Z = 9.381)	Point(X = 296.673, Y = 50.794, Z = 9.390)	Point(X = 296.227, Y = 49.813, Z = 8.565)	Point(X = 298.677, Y = 50.194, Z = 8.551)
pencere	Point(X = 298.677, Y = 50.194, Z = 8.551)	Point(X = 296.227, Y = 49.813, Z = 8.565)	Point(X = 295.820, Y = 48.846, Z = 7.606)	Point(X = 298.020, Y = 49.506, Z = 7.583)
.....	.....	.....	.....	.....

Şekil 17: Revit tabanına eklenmiş parametreler.

### 3. DEĞERLENDİRME

Proje süreci açısından baktığımızda; ön tasarım bölümünde Dynamo form yaklaşımları açısından denemeler yapılan bir GPD olarak kullanılırken, şematik tasarım aşamasından itibaren YBM elemanlarının yönetildiği bir ortam olarak kullanılmıştır (Şekil 18). Süreçte herhangi bir kopukluk olmadan tüm YBM elemanları GPD üzerinden veri tabanına eklenmeye devam etmiştir (Şekil 18). Özellikle YBM merkezli olmayan süreçlerde ortaya çıkan veri kayıpları ya da ortamlar arası iletişim problemleri tamamen aşılmıştır.

Araçların nitelikleri açısından incelendiğinde, ön tasarım bölümünde kullanılan araçların yapabildikleri, büyük ölçüde hatta fazlasıyla, Grasshopper ve eklentileri tarafından da karşılanmaktadır. Tasarım geliştirme aşamasına geçildiğinde Dynamo ortamının avantajları ve “Bilgisayımli YBM” özellikleri daha fazla ön plana çıkmaktadır.



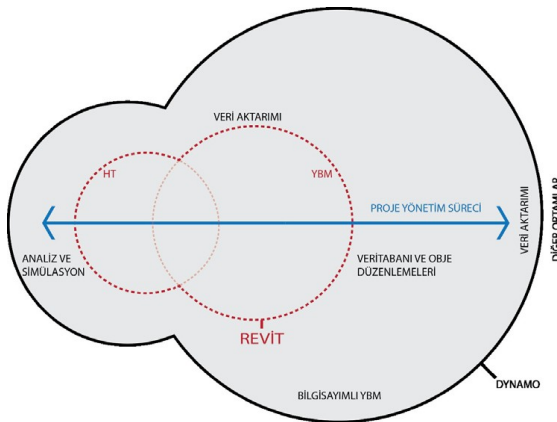
**Şekil 18:** Tasarım modeli proje süreci akış diyagramı.

Dynamo katı modelde değişiklik yapmaktan bir adım öteye geçerek, YBM üzerinde değişiklik yapmaya izin vermektedir. Bu değişiklikler sadece objeler üzerinde değil, Revit ortamındaki obje parametreleri üzerinde de olabilmektedir. Bu sonucu doğuran şey ise temelinde grafik programlama dili olan “Dynamo’ ya” eklenmiş olan Revit sorgulama araçlarıdır. Bu sayede tamamen kullanıcı amaç ve isteği ile şekillenen özelleşmiş uygulamalar oluşturulabilmektedir. Uygulama ekiplerinde ve ofislerde oluşturulan betikler aynı ofis içerisindeki farklı projelerde ve hatta paylaşılması halinde farklı yerlerde de kullanılabilir. Uzun süredir bu konuda oluşturduğum betikleri [www.parametrically.com](http://www.parametrically.com) altında ücretsiz şekilde paylaşmaktayım.

Geliştirilme şekli açısından bakıldığında, Grasshopper gibi Dynamo da açık kaynak olarak geliştirilmektedir ve gönüllü geliştiriciler ve Autodesk geliştiricileri tarafından sürekli katkılar yapılmaktadır. Bütünleşmiş paket yöneticisi kullanılarak, kullanıcılar kendi oluşturdukları araçları diğer kullanıcıların kullanımına sunmaktadır. Tez içerisinde oluşturulan araçlar aynı şekilde paket yöneticisi içerisine “ParametricAlly” adı altında eklenmiştir.

Dynamo, tasarım modeli üzerinden yapılan çalışmalarda YBM platformuyla hesaplamalı tasarım süreçlerini bütünleştirmenin ötesine geçerek, YBM sürecinin kendisini de betikler yardımıyla yönetilebilir ve otomasyon haline dönüştürebilir hale getirmiştir. Dynamo kullanımı öncesinde de kod yazarak veya eklentilerle, tekil süreçler yönetilebilmekteydi; fakat Dynamo hem sürece bütün kullanıcılar tarafından erişilmesini sağlamıştır hem de her türlü süreç için bir grafik programlama alt yapısı sağlamıştır. Bu yüzden bütün süreç “Bilgisayımli YBM” olarak adlandırılmaktadır.

Şekil 19’da anlatılan bütünleşik platform, YBM’nin tutarlı ve tek bir veri tabanından çalışma avantajlarını, hesaplamalı tasarımın ön tasarım ve sematik tasarım evrelerinde sağlamış olduğu esneklik ve deneysel yaklaşım avantajları ile birleştirmektedir. Veri aktarma işlemi, geleneksel çizim temsili şeklinde olduğunda Revit’in paftalama araçları kullanılırken, Revit’in ön tanımlı araçlarının yetersiz kaldığı veya sayısal bir verinin ayıklanarak aktarım yapılması gereken durumlarda da Dynamo kullanılabilir. Hesaplamalı Tasarımdan (HT) YBM’ye geçiş Şekil 19’da iki paradigmanın kesiştiği aralık olarak temsil edilmiştir ve bu kesişim aslında aynı ortamda devam ettiği için net sınırları olmayan bir hal almıştır.



Şekil 19: Dynamo ile oluşturulan tasarım modeli. Kaynak referans alınarak yeniden yorumlanmıştır (Humppi & Österlund, 2011).

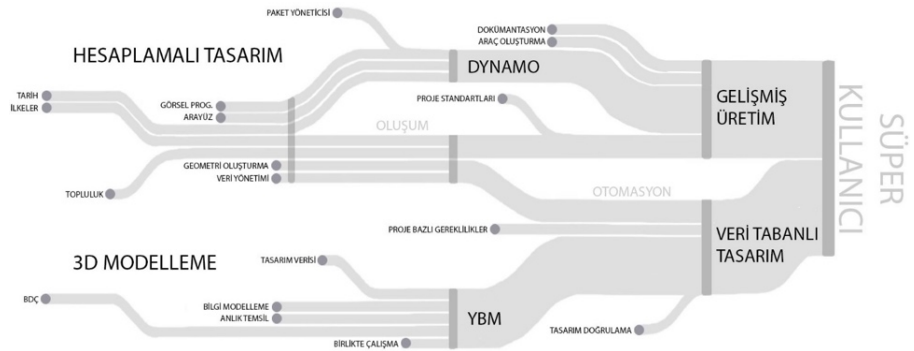
YBM ve HT entegrasyonunun Mimarlık Mühendislik ve İnşaat sektörü etkileri üzerine yapılan bir araştırmada, bu entegrasyonun problemlerin üstesinden gelmede ne kadar yardımcı olabileceği hakkındaki soruya, sektör çalışanlarının %78.9'u, öğrencilerin %64.7'si ve araştırmacıların %50'si olumlu cevap vermiştir (Abrishami et al., 2014). **Tablo 1** incelendiğinde de çalışanlarda, öğrencilerde ve araştırmacılar da çok yüksek bir çoğunlukla HT ve YBM entegrasyonu sonucunda YBM'nin de kapasitesinin artacağına ilişkin beklenti oldukça fazladır.

		HT, kullanıcıların tasarım kapasitesini arttırabilir mi?			HT, YBM kapasitesini arttırabilir mi?		
		Emin Değilim	Hayır	Evet	Emin Değilim	Hayır	Evet
Gruplar	Çalışan	13.2%	0%	86.8%	15.8%	0.0%	84.2%
	Öğrenci	11.8%	5.9%	82.4%	11.8%	5.9%	82.4%
	Araştırmacı	20%	6.7%	73.3%	33.3%	6.7%	60.0%
Toplam		<b>20.0%</b>	<b>6.7%</b>	<b>81.7%</b>	<b>17.0%</b>	<b>4.6%</b>	<b>78.4%</b>

**Tablo 1:** HT kullanıcı kapasitesi ve YBM kapasitesi üzerine etkileri (Abrishami et al., 2014).

Özellikle Dynamo'nun YBM ortamını çevreleyen bir yapı olduğu düşünüldüğünde, Bilgisayar destekli tasarımın gelişimindeki paralel iki yönün bugün tek bir ortamda birleşebildiğini görmekteyiz. **Şekil 20**'de farklı özellikleri kullanabildiğinden bahsedilen kurgusal süper kullanıcı bu özelliklerin birçoğunu Dynamo ve Revit birlikteliğinde toplayabilmektedir.

**Şekil 20:** YBM ve HT kullanımını harmanlayan kullanıcı (Deutsch, 2017).



## 4. SONUÇLAR

Bilgisayarın mimari tasarım ortamı olarak kullanımının iki yönü olan YBM ve HT sürekli olarak birlikte kullanılmaya çalışılmış ve entegrasyonu için farklı yöntemler denendiği literatur araştırmaları sırasında görülmüştür. Özellikle GH'in gelişmiş bir HT ortamı olmasından dolayı iş akışlarında merkezde tercih edilmiştir. Gelişen bilgi sistemlerinden dolayı YBM'nin kullanımının artmasıyla birlikte entegrasyonun önemi daha da artmış, mevcut yöntemler eksik kalmaya başlamıştır ve YBM ile entegre olmuş bir GPD'nin gerekliliği daha da artmıştır. Bu entegrasyonu sağlamak adına kullanılan araçların ya sadece geometri üzerinden aktarımlar yaptığı ya da YBM ortamında sadece belirli düzeylerde kontrole izin verdiği görülmüştür. Dynamo bu noktada aynı amaçlarla kullanılan araçların aksine YBM üzerinde çalıştığı için öne çıkmıştır. DesignScript gibi gelişmiş bir başka dilin Dynamo ortamıyla birleştirilmesiyle güçlü bir altyapı kullanıma sunulmuştur.

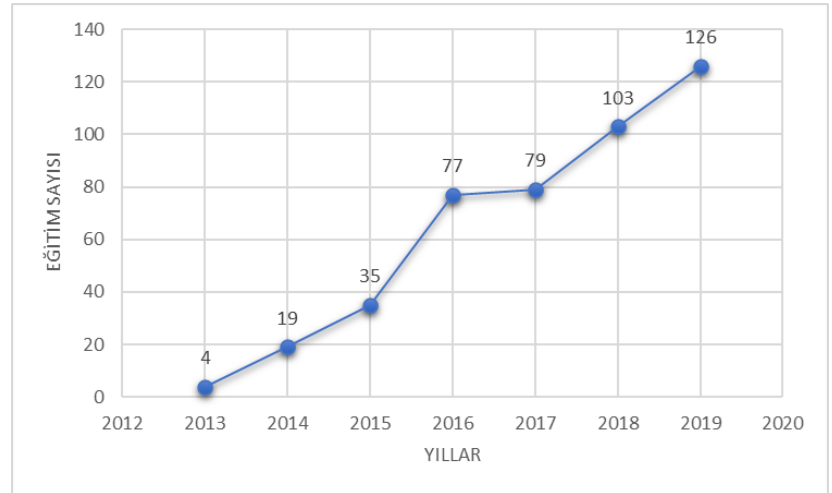
GPD ile oluşturulan entegrasyonun B-YBM (Bilgisayimli YBM) olarak yeni bir kullanım alanı açtığı tasarım modeli üzerinden görülmüştür. Tasarımın ön safhalarında GH benzeri özellikleri için kullanılan Dynamo, proje süreci içerisinde YBM veri tabanı ve değişkenleri üzerinde kontrol sağlayan betiklerle B-YBM aracı olarak kullanılmıştır. Dynamo, hesaplamalı tasarım araçları tarafında hala en çok kullanılan araç olan Grasshopper'ın, eklenti ve kullanıcı desteği anlamında gerisinde olmasına rağmen Dynamo'nun da GH gibi açık kaynak olması, büyüyen bir topluluğun oluşmasını sağlamaktadır, öyle ki Grasshopper'a eklentiler yazan programcılar benzerlerini Dynamo ortamına aktarmaktadır (Elk, FireFly, LunchBox, LadyBug vs.). 'Refinery' gibi opsiyon denemeleri yapabilen yapıların Autodesk tarafından farklı araçlarla desteklendiği takdirde daha komple bir B-YBM altyapısı kullanıcılara sunulabilir.

Paket yöneticisi sayesinde kullanıcılar arasında "node" paylaşımı yapılabilmektedir. Birçok kullanıcı kendi "node" gruplarını oluşturup bu yönetici içerisinde paylaşıma sunmuştur. Bu yönüyle giriş seviyesindeki kullanıcılar için hızlı bir kullanma ve öğrenme süreci oluşmaktadır. DesignScript'in özellikleri arasında yer alan "NodeToCode" özelliği de bu öğretici yapıyı pekiştirmektedir. Aynı şekilde uzman kullanıcılar için de paylaşılan bu araçlar iş akışı içerisinde kolaylıklar ve iş gücü kazancı sağlamaktadır.

YBM ortamının otomasyonu açısından ele alındığında ise Dynamo ortamı benzersiz bir yapıdır ve sağladığı en büyük avantajı da şüphesiz bu yönüdür. GPD ve YBM entegrasyonu bu aşamada sadece tasarım yönüne etki etmemiştir ve YBM veri tabanı ve objeleri GPD ile düzenlenebilir hale gelmiştir. Kurulan mantıksal ve matematiksel betikler sayesinde oluşturulan otomasyonlar hem iş gücünden kazanç hem de proje ortamında bir standartlaşma oluşturmayı sağlamışlardır.

**Şekil 21**'de yıllara göre dersler eğrisine bakıldığında ise; "Autodesk University" sayfasında Dynamo başlıklı ders sayısı 2013 yılında dört, 2014 yılında 19, 2015 yılında bu sayı otuz beş, 2016 yılında ise yetmiş yedi, 2017 yılında yetmiş dokuz, 2018 yılında yüz üç, 2019 yılında da yüz yirmi altı olmuştur.

Türkiye'de de 2015 yılı içerisinde yapılan iki farklı çalıştayda toplam iki yüze yakın katılım sağlanmıştır. İstanbul Teknik Üniversitesi FBE Bilişim Anabilim Dalı Mimari Tasarımda Bilişim Lisansüstü Programı ve KSB Hesaplamalı Tasarım Kulübü bünyesinde 2016-17 güz yarıyılında aylık olarak kullanıcı toplantıları gerçekleştirilmiş ve bilgi paylaşımı yapılması sağlanmıştır. Bu açıdan bakıldığında da Dynamo kullanımı ve desteği ülke içinde ve dışında yükselen bir eğri ile artmaktadır.



**Şekil 21:** Autodesk University  
Yıllara göre dersler.

## Kaynakça

Abrishami, S., Goulding, J. S., Pour Rahimian, F., Ganah, A., & Sawhney, A. (2014). G-BIM Framework: A Feasibility Study for the Adoption of Generative BIM Workspace for Conceptual Design Automation. *Creative Construction Conference 2014*, 539–543.

Deutsch, R. (2017). *Convergence: The Redesign of Design* (1st ed.). Wiley.

Garber, R. (2014). BIM Design: Realising the Creative Potential of Building Information Modelling. In *BIM Design: Realising the Creative Potential of Building Information Modelling*. <https://doi.org/10.1002/9781118914694>

Humppi, H., & Österlund, T. (2011). Algorithm-Aided BIM. *2*, 601–609.