

Sistem Modelleme

- Sistem modelleme, bir sistemin soyut modellerinin (UML-Birleşik Modelleme Dili yada Biçimsel (Matematiksel) Modeller) geliştirilmesi sürecidir.
- Her bir soyut model sistemin farklı bir görünümünü veya perspektifini sunar.
- Gereksinim mühendisliği sürecinde sistemin detaylı gereksinimleri saptanır.
- Modeller, tasarım sürecinde sistemi geliştiren mühendislerle sistemi betimlemek amacıyla ve geliştirme sonrasında ise sistemin yapısını ve işlevini belgelemek amacıyla kullanılır.

- Yazılım sektöründe modelleme için geliştirilmiş EXPRESS, EXPRESS-G, IDEF, UML gibi çeşitli modelleme dilleri mevcut olup, nesne tabanlı sistemlerin incelenmesi ve tasarımında standart olarak kullanılan modelleme dili UML'dir (Unified Modeling Language - Birleşik Modelleme Dili).
- Yazılımda sistem modelleme süreci problemin tanımının yapılması ve bu probleme bir çözüm oluşturulması aşamalarını içerir.
- UML modelleri yalnızca yazılım profesyonelleri tarafından kullanılmakla kalmayıp, otomatik kod üreticileri tarafından da okunabilme özelliğine sahiptir.

- Sistem değişik açılardan modellenenebilir;

Dış perspektif

Sistemin bağlam veya çevresini modelleme

İç Perspektif

Sistemin çevresi ile olan etkileşimini veya sistemin bileşenleri arasındaki etkileşimi modelleme

Yapısal perspektif

Sistemin organizasyonunu veya sistem tarafından işlenen verinin yapısını modelleme

Davranışsal perspektif

Sistemin dinamik davranışını ve olaylara tepkisini modelleme

- Grafiksel modellerin üç yaygın kullanımı vardır;
- Var olan veya önerilen sistem üzerinde tartışmayı tetiklemek ve odaklanmanın bir yolu olarak (Özellikle çevik modellemede).
- Var olan bir sistemi belgelemenin bir yolu olarak.
- Sistemin bir gerçekleştirimini üretmek için kullanılacak detaylı bir sistem tanımı olarak.

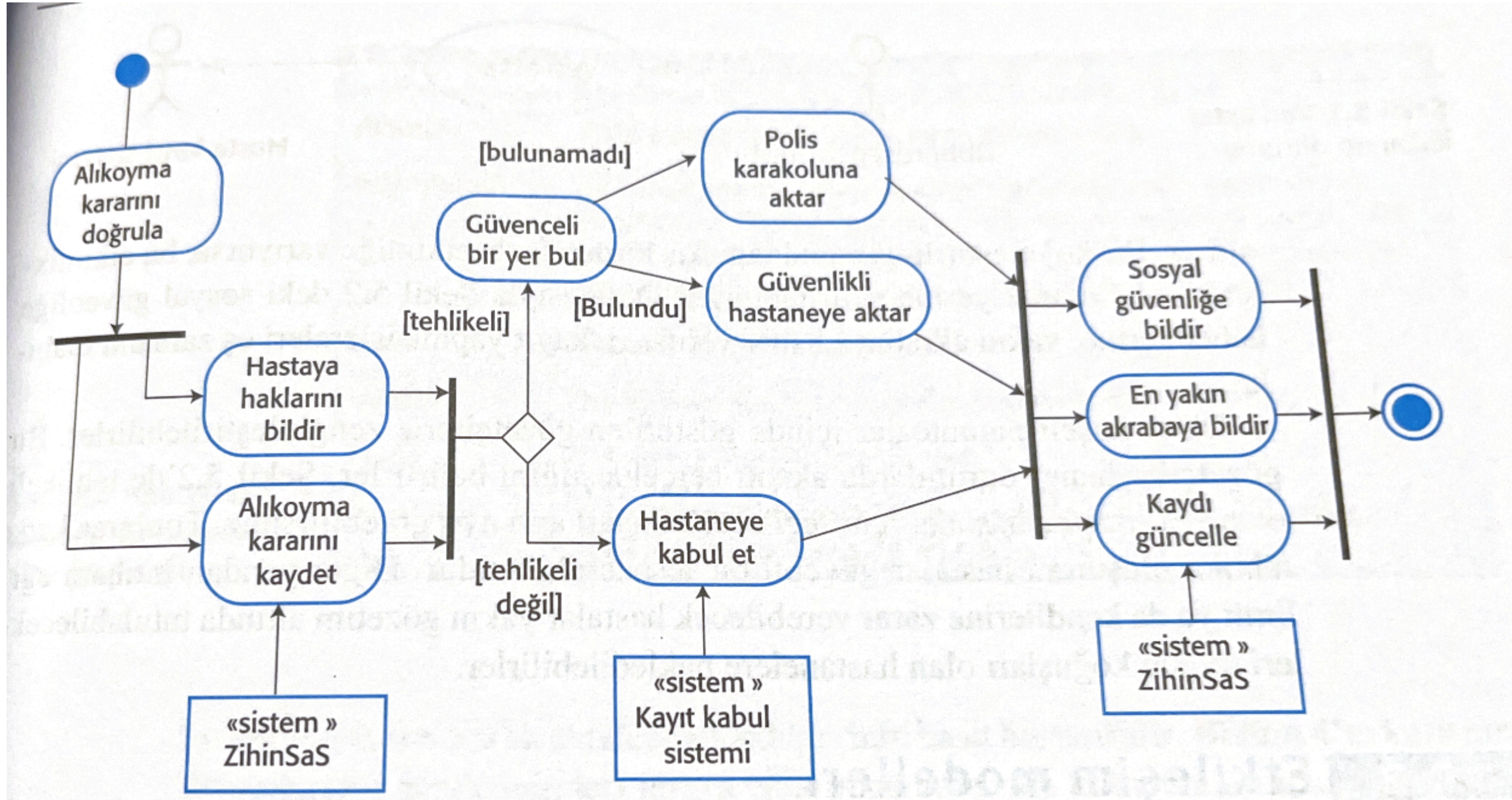
UML (Birleşik Modelleme Dili)

- 13 diyagram tipine sahiptir. Ancak 5 tanesi aktif kullanılmaktadır.
 1. Etkinlik (Activity) Diyagramları: Süreç veya veri işleme görevi içindeki etkinlikleri gösterirler. Sistemin kullanıldığı iş süreçlerini gösterebilir.
 2. Kullanım Durumu (Use Case) Diyagramları: Sistem ile çevresi arasındaki etkileşimi gösterirler.
 3. Sıra (Sequence) Diyagramları: Aktörler ve sistem arasındaki etkileşimlerle sistem bileşenleri arasındaki etkileşimleri gösterirler.
 4. Sınıf (Class) Diyagramları: Nesne sınıfları ile bunlar arasındaki bağlantıları gösterirler.
 5. Durum (State) Diyagramları: Sistemin iç ve dış olaylara nasıl tepki verdiğini gösterirler.

Etkinlik (Activity) Diyagramları

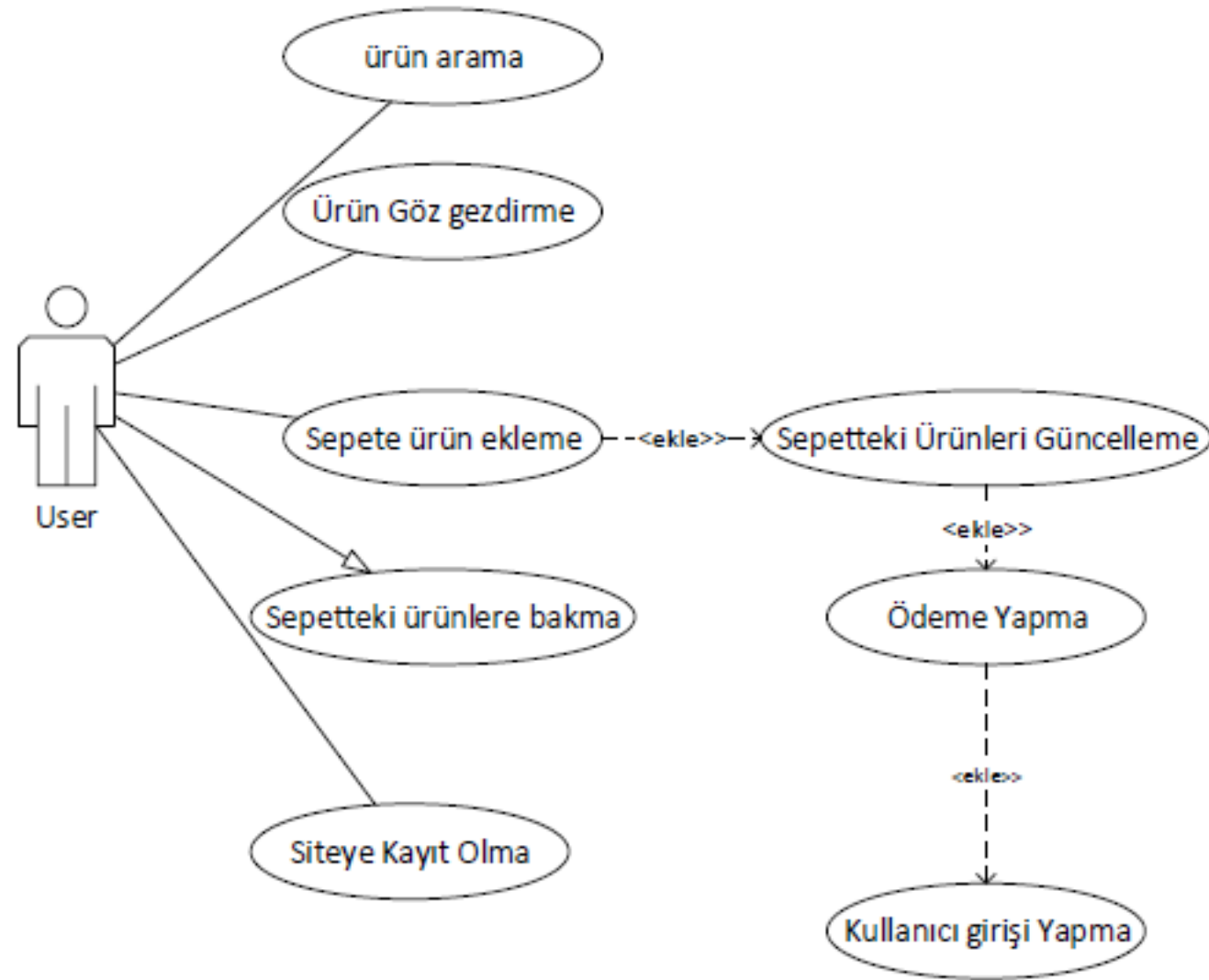
- Sistemin akış yönünden davranışını gösterir.
- Durum diyagramları gerçekleşen eylemler neticesinde oluşan durumları nitelerken etkinlik diyagramları ise koşullu ya da paralel ilerleyen eylemleri betimlemesi açısından birbirinden farklıdır.

UML Etkinlik Diyagramı Örnek



Kullanım Durumu (Use Case) Diyagramları

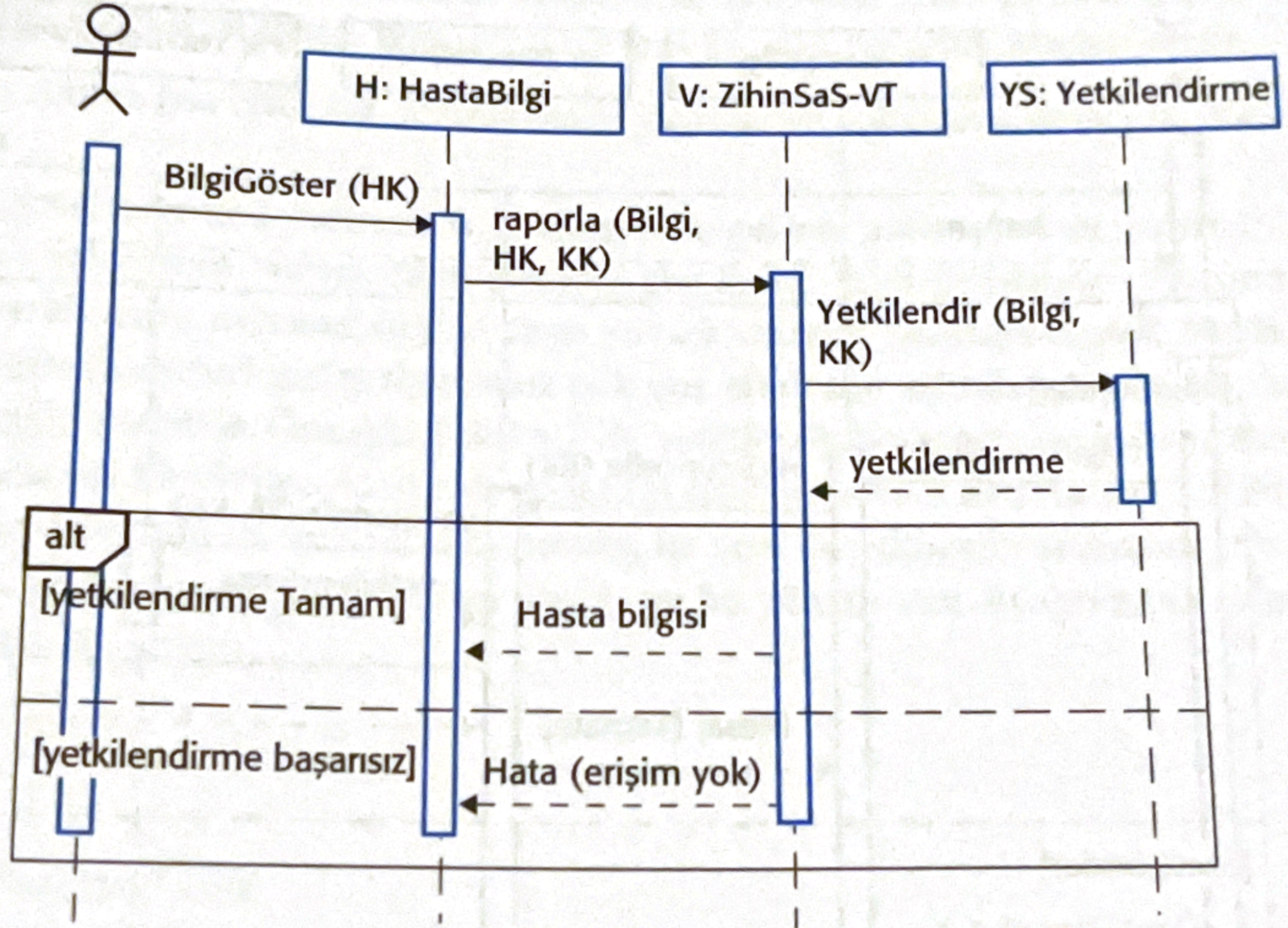
- Süreçler yönetilirken ihtiyaç duyulan tüm fonksiyonları, bu fonksiyonları tetikleyecek aktörleri, fonksiyonlardan etkilenecek aktörleri ve fonksiyonlar arasındaki ilişkileri göstermek amacıyla kullanılmaktadır.
- Diyagram, bir uygulamanın sistemini/alt sistemini modellemek için kullanılır.



Sıra (Sequence) Diyagramları

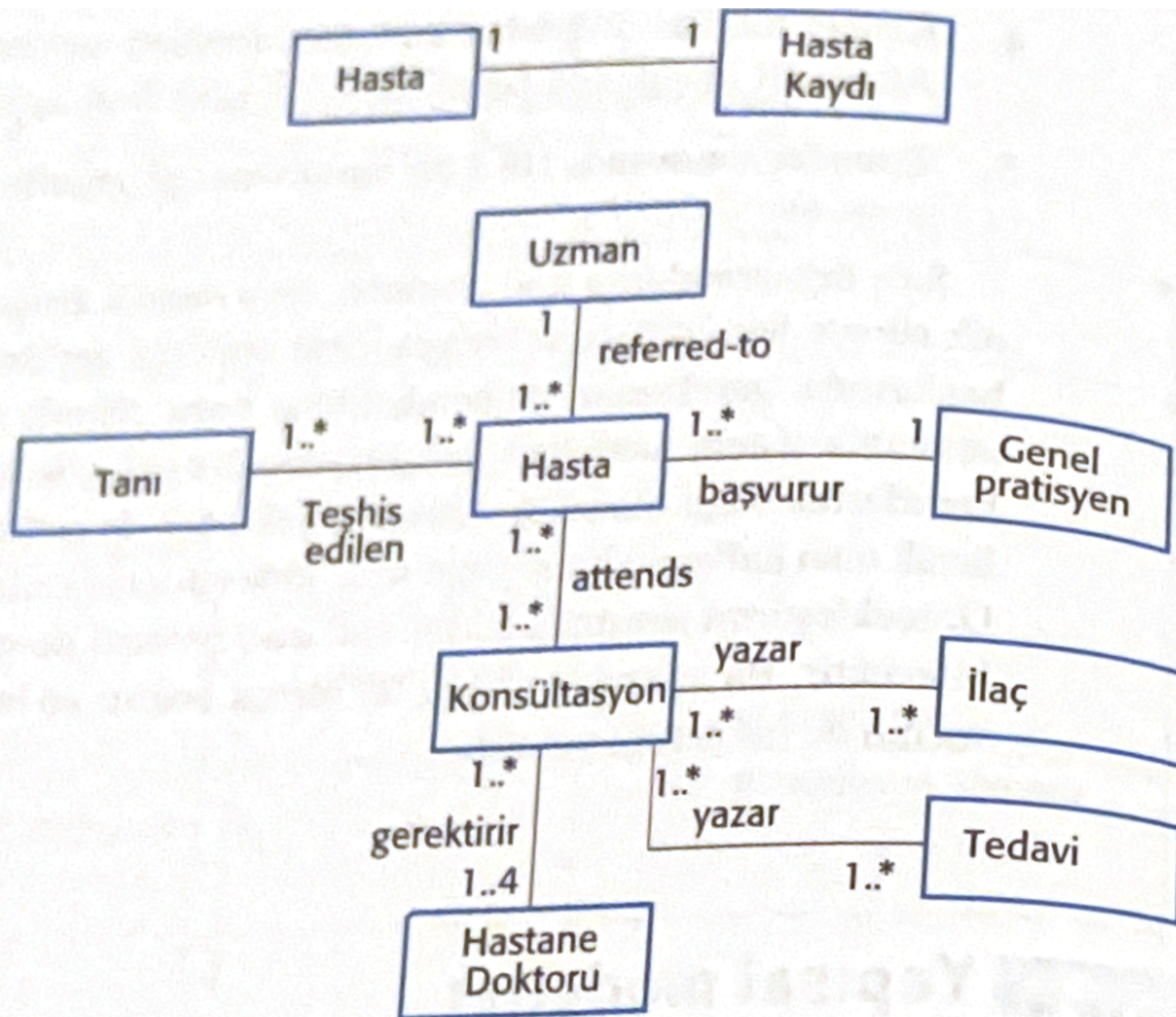
- Bir sistemin kullanıcıları ile nesneleri arasında ya da nesnelerin kendi aralarındaki etkileşimleri modellemek amacıyla kullanılır.
- Mesajlar soldan sağa olacak şekilde düzenlenince sistem içindeki sıralı veri işleme zincirini gösterir.
- Veri işleme modelleri daha anlaşılır olduğu için paydaşlar tarafından tercih edilse de mühendisler sıralı diyagramları tercih etmektedir.

Tıbbi resepsiyonist



Sınıf (Class) Diyagramları

- Nesneye yönelik sistem modeli geliştirilirken sınıfları ve sınıflar arasındaki bağları gösterirler.



Konsültasyon

Doktorlar

Tarih

Saat

Klinik

Sebep

...

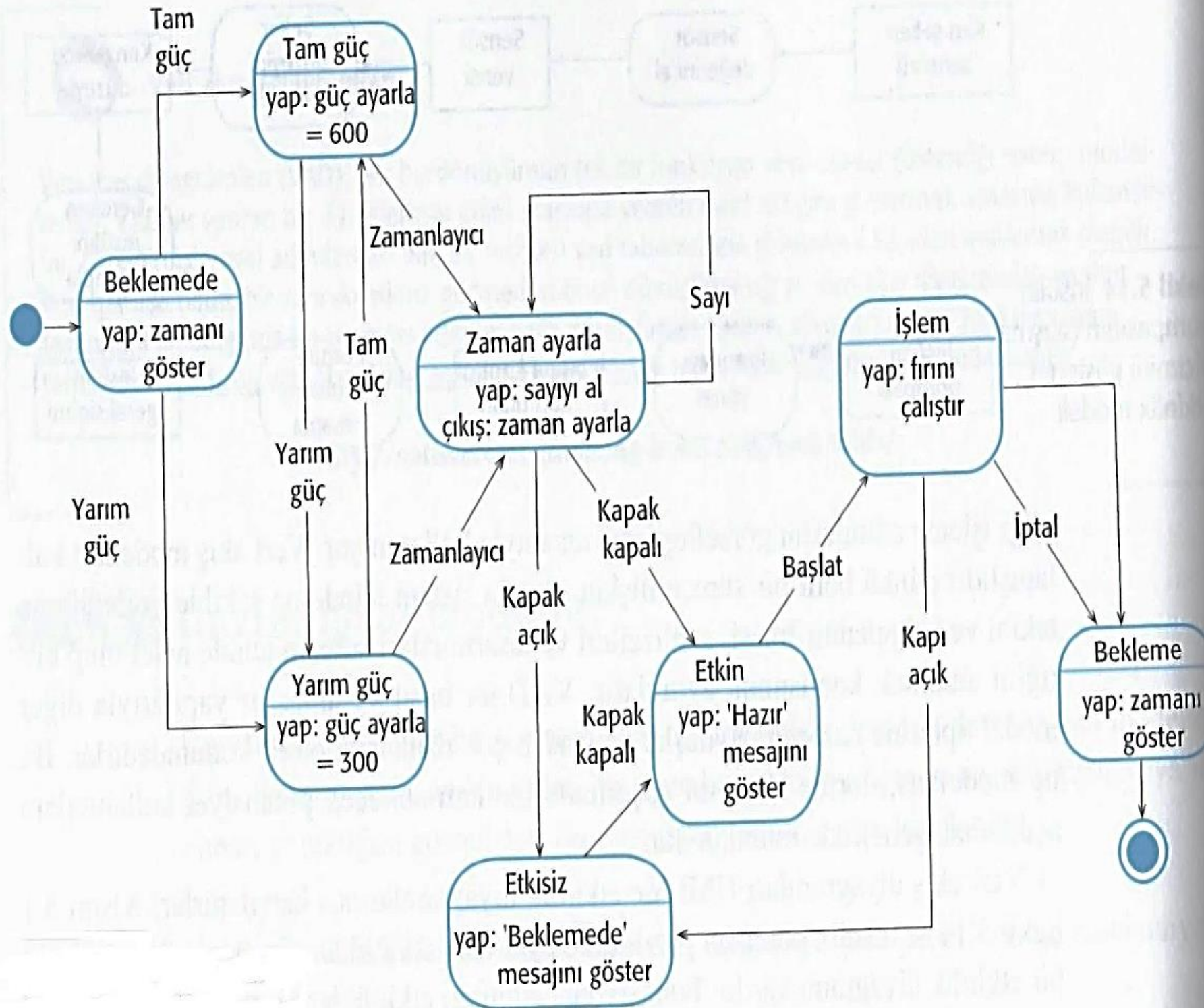
Yeni()

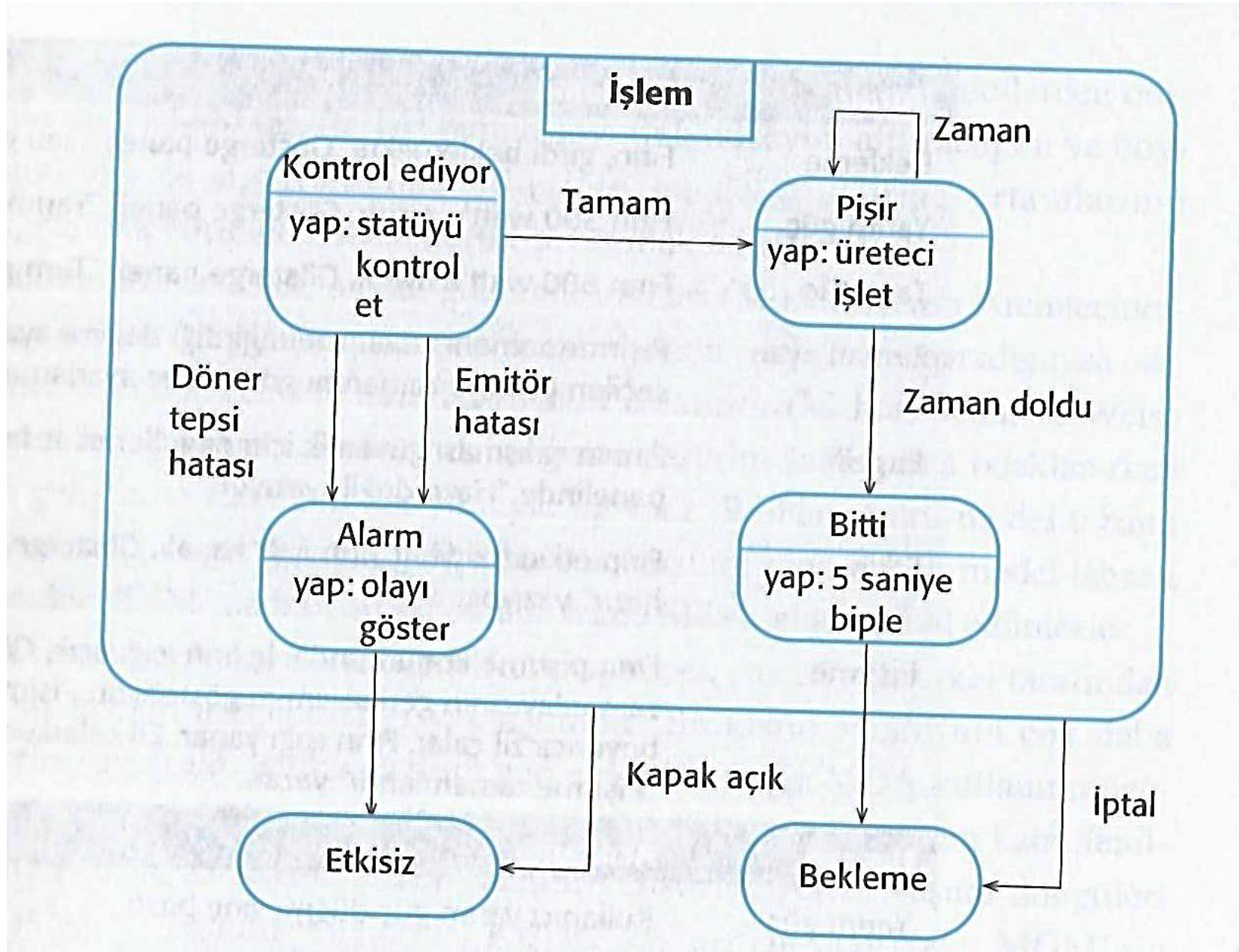
ReçeteYaz()

...

Durum (State) Diyagramları

- Sistemin davranışlarını modeller.
- Bir olay gerçekleştiğinde olası tüm durumları tanımlar.
- Az sayıda eleman içerir. Temel elemanlar, nesnenin durumunu temsil eden kutular ve bir sonraki duruma geçişi temsil eden oklardır. Eylem kısmı nesnenin o durumda ne yapacağını gösterir.



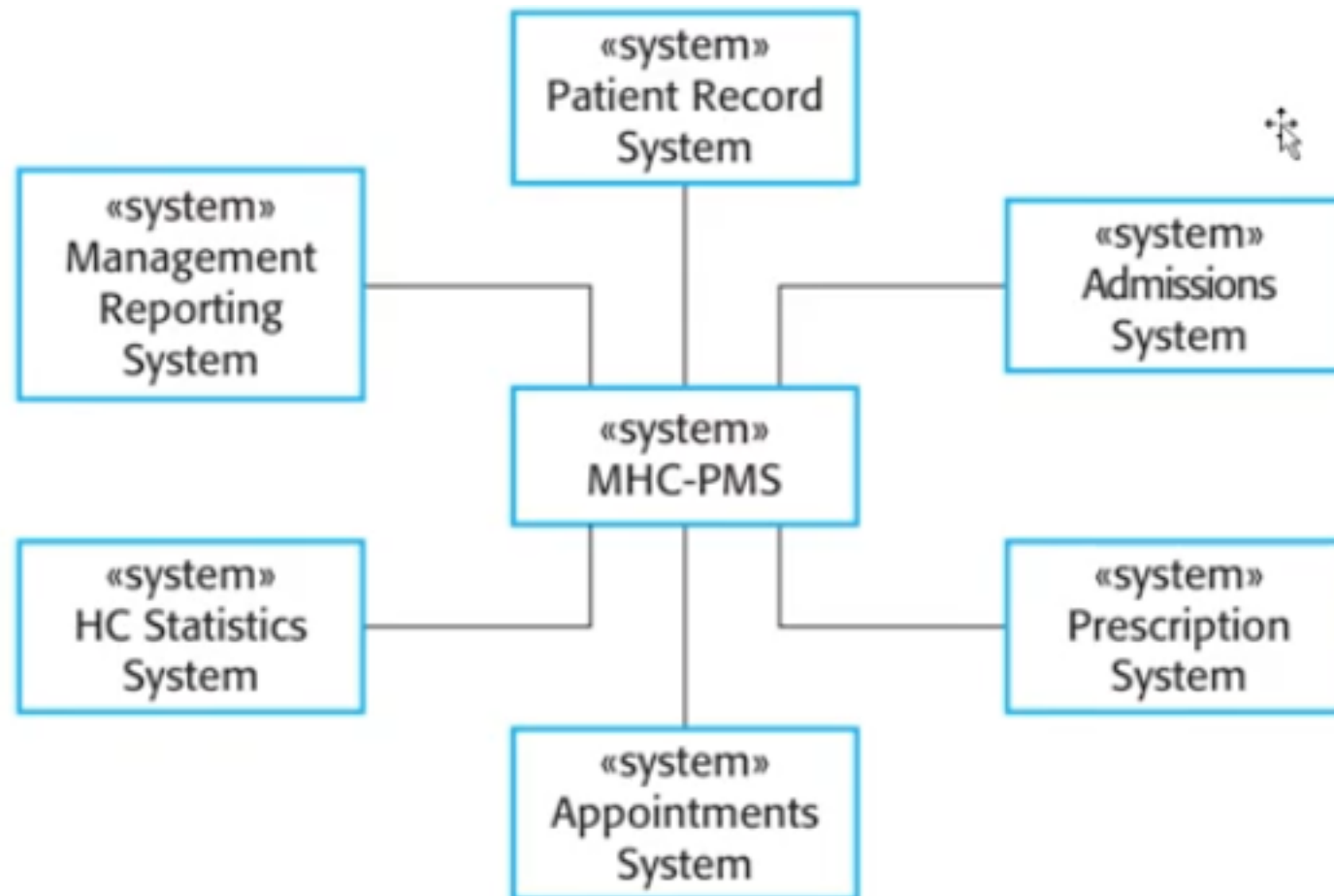


Sistemi Değişik Açılardan Modellemek

- Bağlam modelleri: Sistemin bağlam veya çerçevesini modellediğimiz dış perspektif.
- Etkileşim modelleri: Sistemin çevresi ile veya sistemin bileşenlerinin birbiri ile olan etkileşimini modellediğimiz iç perspektif.
- Yapısal modeller: Sistemin organizasyonunu veya sistem tarafından işlenen verinin yapısını modellediğimiz yapısal perspektif.
- Davranışsal modeller: Sistemin dinamik davranışlarını ve olaylara tepkisini modellediğimiz davranışsal perspektif.

Bağlam Modelleri

- Sistemi tasarlamadan önce neyin sistemin içinde neyin dışında kalacağına karar vermek gerekir.
- Paydaşlar ile sistemin hangi işlevleri içereceğine karar verilir. Operasyonel çevrede sistemin işleme ve çalışma biçimleri belirlenir.
- Ortamdaki pek çok başka otomasyon sistemini kapsayabilir fakat betimlenen sistem ile çevredeki diğer sistemler arasındaki ilişkileri göstermez.
- UML Etkinlik Diyagramları.



Etkileşim Modelleri

- Tüm sistemler, kullanıcı girdi-çıkıtlı şeklindeki kullanıcı etkileşimi veya diğer sistemler arasında ve sistemin bileşenleri arasında etkileşim gerektirir.
- Bu modelleme sayesinde kullanıcı gereksinimleri belirlenir ve sistemler arasındaki modelleme sayesinde iletişim problemleri, sistem performansı ve güvenlik gereksinimleri ortaya çıkar.
- Etkileşim modelleme yaklaşımları;
 1. UML Kullanım durumu modelleme
 2. UML Sıra diyagramları

Kullanım Durumu Modelleme

- 1990'larda Ivar Jacobsen tarafından geliştirilmiş.
- Kullanım durumu, kullanıcının belli bir etkileşim sonucu sistemden beklentilerinin basit bir tarifidir.
- Sistem tasarımının ilk evrelerinde daha kullanışlıdır.

Kullanıcılar

Öğrenci işleri personeli, OBİS

Betimleme

Personel OBİS sistemindeki verileri Yöksis veri tabanına aktarabilir.
Aktarılan veri kişinin mezuniyet durumu, diploma notu vb. olabilir.

Veri

Öğrencinin kimlik bilgileri, transkripti

Uyaran

Personel tarafından verilen kullanıcı komutu

Tepki

Yöksis'in güncellendiği teyidi

Yorumlar

Personel öğrenci bilgisine ve Yöksis'e erişmek için gerekli güvenlik izinlerine sahip olmalıdır

Yapısal Modeller

- Bir sistemin organizasyonunu o sistemi oluşturan bileşenler ve bileşenler arasındaki ilişkiler cinsinden gösterir.
- Sistem tasarımını gösteren statik modeller olabileceği gibi sistem çalışırken organizasyonu gösteren dinamik modeller de olabilir.
- Sistem mimarisi tasarlanırken oluşturulur.
- UML Sınıf Diyagramları.

Davranışsal Modeller

- Bir sistemin işleyişi sırasındaki dinamik davranışının modelleridir.
- Sistemin işleyişini tetikleyecek bir olay ve ya veri girişi meydana gelir.

- Veri G d ml  Modelleme;
- Girdi veriyi i lemek ve gereken  ıktıyı  retmek i in gereken etkinlikleri tanımlamak i in kullanılır.
- Payda lar tarafından daha anlaşılır modeldir.

- Olay G d ml  Modelleme;
 - Bir sistemin i  ve dı  olaylara nasıl tepki verdiđini g stermeyi sađlamaktadır.
 - Sistem sınırlı sayıda duruma sahip olmalı ve olayların bir durumda diđerine ge i e yol a acađı varsayılmaktadır.
 - Ger ek zamanlı sistemlerin tasarımı ve belgelendirilmesinde sık a kullanılmaktadır.
 - UML Durum diyagramları ile temsil edilir.

- Durum tabanlı modellemede sorun, mümkün durum sayısının hızla artmasıdır. Bu sebeple detayların gizlenmesi gerekmektedir.
- Bu sebeple “süper durum” kullanılır. Üst modelde tek bir durum gibi gözükse de daha fazla detay gösterilmek istendiğinde ayrı bir diyagrama dönüşecek biçimde genişleyebilir.

- Model Güdümlü Mühendislik;
 - Ana çıktısı programlar yerine modeller olan geliştirme sürecini içeren bir yaklaşımdır.
 - Bir donanım / yazılım platformunda çalışan programlar daha sonra modellerden otomatik olarak oluşturulur.
 - Yazılım mühendisliğinde soyutlama düzeyini yükselttiği, böylece mühendislerin artık programlama dili ayrıntılarıyla veya yürütme platformlarının özellikleriyle ilgilenmelerine gerek kalmadığı düşünülmekte.
 - Model Güdümlü Mimari fikrinde gelişmiştir.

Model G d ml  Mimari

- Model g d ml  mimari, model g d ml  m hendisliđin  nc s yd .
- Bir sistemi tanımlamak i in UML modellerinin bir alt k mesini kullanan yazılım tasarımı ve uygulamasına y nelik model odaklı bir yaklaşımdır.
- Farklı soyutlama seviyelerinde modeller yaratılır. Y ksek seviyeli, platformdan bağımsız bir modelden, prensip olarak, insan m dahalesi olmaksızın bir  alıřma programı oluřturmak m mk nd r.

1. Bilgisayardan Bağımsız Model (Computation Independent Model - CIM): Bunlar, bir sistemde kullanılan önemli alan soyutlamalarını modeller. Bir sistemin farklı görünümelerini yansıtmak için birden fazla geliştirilebilir. CIM'lere bazen etki alanı modelleri de denir.
2. Platformdan Bağımsız Model (Platform Independent Model - PIM): Genellikle statik sistem yapısını ve bu yapının dış ve iç olaylara nasıl tepki verdiğini gösteren UML modelleri kullanılarak tanımlanır.
3. Platforma Özel Modeller (Platform Specific Models - PSM): Bir uygulama platformu için ayrı bir PSM olacak biçimde PIM'lerin dönüştürülmesi ile elde edilir.

- Model Tabanlı Mühendislik yüksek soyutlama düzeyinde olması sebebiyle, tasarım ve gerçekleştirim sürelerini hızlandırır, hatayı azaltır, yeniden kullanılabilir, platformdan bağımsız uygulama modellerinin oluşturmaya olanak sağlar.

Kaynakça

- Ian Sommerville, Yazılım Mühendisliği
- <https://medium.com/@veysel.gunes36/use-case-kullanım-şekli-diyagramları-ile-resmin-bütünü-görmek>
- Doç. Dr. Sedat AKLEYLEK'in ders notları