

# IT522 – Yazılım Mühendisliği 2021



PhD Furkan Gözükkara, Toros University

<https://github.com/FurkanGozukara/Yazilim-Muhendisligi-IT522-2021>

## Ders 5

# Sistem Modelleme



Kaynak : <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Presentations/index.html>

# Ders 5 İşlenmiş Konular

---



- Bağlam modelleri
- Etkileşim modelleri
- Yapısal modeller
- Davranış modelleri
- Model odaklı mühendislik

# Sistem Modelleme

---



- Sistem modelleme, her modelin o sistemin farklı bir görünümünü veya perspektifini sunduğu bir sistemin soyut modellerini geliştirme sürecidir.
- Sistem modelleme, şimdi neredeyse her zaman Birleşik Modelleme Dilindeki (UML) gösterimlere dayanan bir tür grafiksel gösterim kullanan bir sistemi temsil etmek anlamına geldi.
- Sistem modelleme, analistin sistemin işlevselliğini anlamasına yardımcı olur ve modeller müşterilerle iletişim kurmak için kullanılır.

# Mevcut ve Planlanan Sistem Modelleri



- İhtiyaç mühendisliği sırasında mevcut sistemin modelleri kullanılır. Mevcut sistemin ne yaptığını netleştirmeye yardımcı olurlar ve güçlü ve zayıf yönlerini tartışmak için bir temel olarak kullanılabilirler. Bunlar daha sonra yeni sistem için gereksinimlere yol açar.
- Yeni sistemin modelleri, önerilen gereksinimleri diğer sistem paydaşlarına açıklamaya yardımcı olmak için gereksinim mühendisliği sırasında kullanılır. Mühendisler, tasarım önerilerini tartışmak ve uygulama için sistemi belgelemek için bu modelleri kullanır.
- Model odaklı bir mühendislik sürecinde, sistem modelinden tam veya kısmi bir sistem uygulaması oluşturmak mümkündür.

# Sistem Perspektifleri

---



- Sistemin bağlamını veya ortamını modellediğiniz harici bir perspektif.
- Bir sistem ile çevresi arasındaki veya bir sistemin bileşenleri arasındaki etkileşimleri modellediğiniz bir etkileşim perspektifi.
- Bir sistemin organizasyonunu veya sistem tarafından işlenen verilerin yapısını modellediğiniz yapısal bir perspektif.
- Sistemin dinamik davranışını ve olaylara nasıl tepki verdiğini modellediğiniz davranışsal bir perspektif.

# UML Diyagram Türleri



- Bir işlemde veya veri işlemede yer alan etkinlikleri gösteren etkinlik diyagramları.
- Bir sistem ile çevresi arasındaki etkileşimleri gösteren durum diyagramlarını kullanın.
- Aktörler ile sistem arasındaki ve sistem bileşenleri arasındaki etkileşimleri gösteren sıra diyagramları.
- Sistemdeki nesne sınıflarını ve bu sınıflar arasındaki ilişkileri gösteren sınıf diyagramları.
- Sistemin iç ve dış olaylara nasıl tepki verdiğini gösteren durum diyagramları.

# Grafik Modellerin Kullanımı



- Mevcut veya önerilen bir sistem hakkında tartışmayı kolaylaştırmanın bir yolu olarak
  - Eksik ve yanlış modeller, rolleri tartışmayı desteklemek olduğu için kabul edilebilir.
- Mevcut bir sistemi belgelemenin bir yolu olarak
  - Modeller, sistemin doğru bir temsili olmalıdır ancak eksiksiz olmaları gerekmez.
- Bir sistem uygulaması oluşturmak için kullanılabilecek ayrıntılı bir sistem açıklaması olarak
  - Modeller hem doğru hem de eksiksiz olmalıdır.

# Bağlam Modelleri



- Bağlam modelleri, bir sistemin operasyonel bağlamını göstermek için kullanılır - sistem sınırlarının dışında ne olduğunu gösterirler.
- Sosyal ve örgütsel kaygılar, sistem sınırlarının nereye konumlandırılacağına ilişkin kararı etkileyebilir.
- Mimari modeller, sistemi ve diğer sistemlerle ilişkisini gösterir.



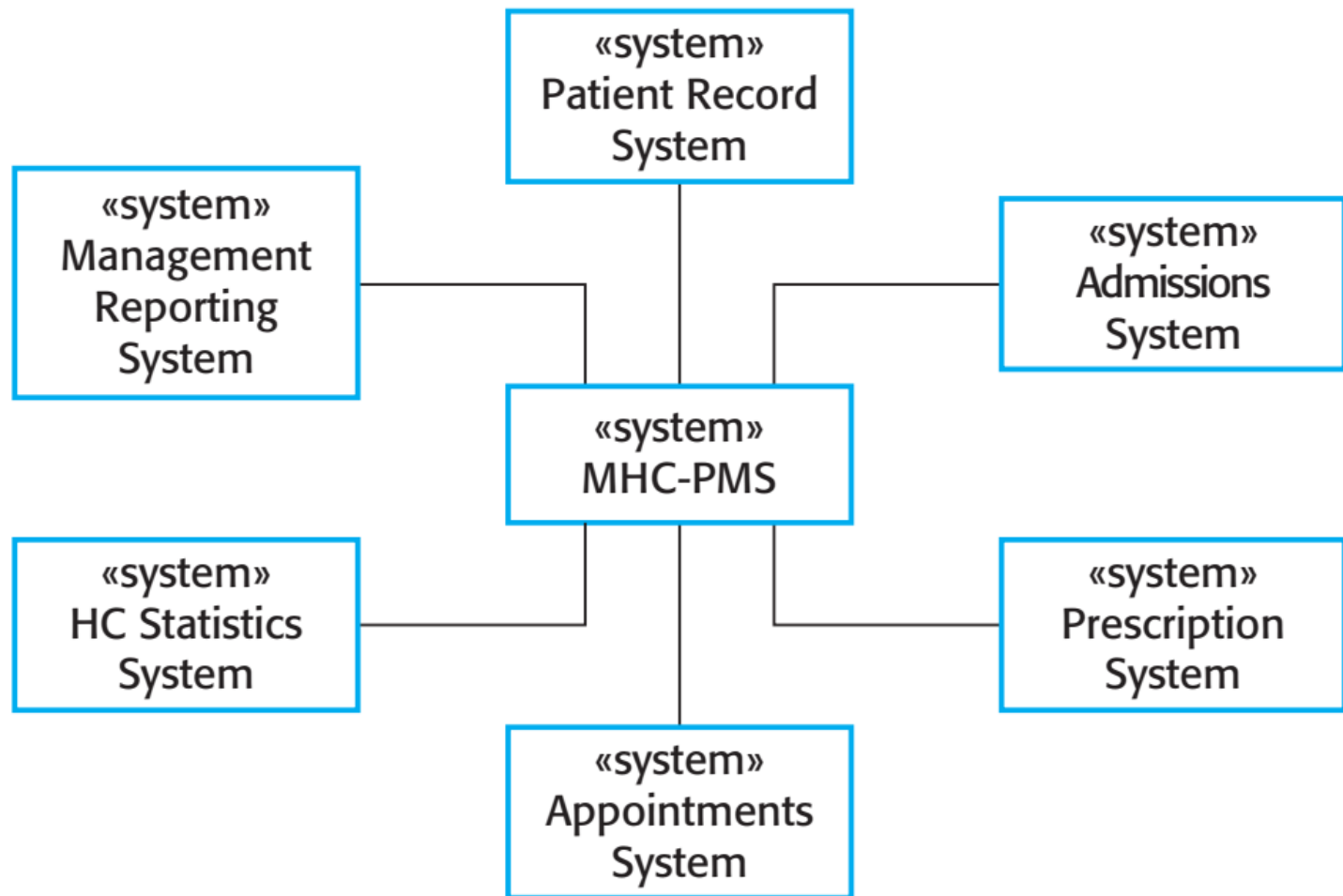
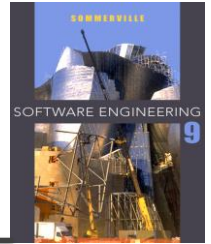
# Sistem Sınırları

---



- Sistem sınırları, sistemin içinde ve dışında olanı tanımlamak için oluşturulur.
  - Kullanılan veya geliştirilmekte olan sisteme bağlı olan diğer sistemleri gösterirler.
- Sistem sınırının konumu, sistem gereksinimleri üzerinde derin bir etkiye sahiptir.
- Bir sistem sınırının belirlenmesi siyasi bir yargıdır
  - Bir kuruluşun farklı bölümlerinin etkisini veya iş yükünü artıran / azaltan sistem sınırları geliştirmek için baskılar olabilir.

# AS-HYS (Akıl Sağlığı Hasta Yönetim Sistemi – AS-HYS) Bağlamı



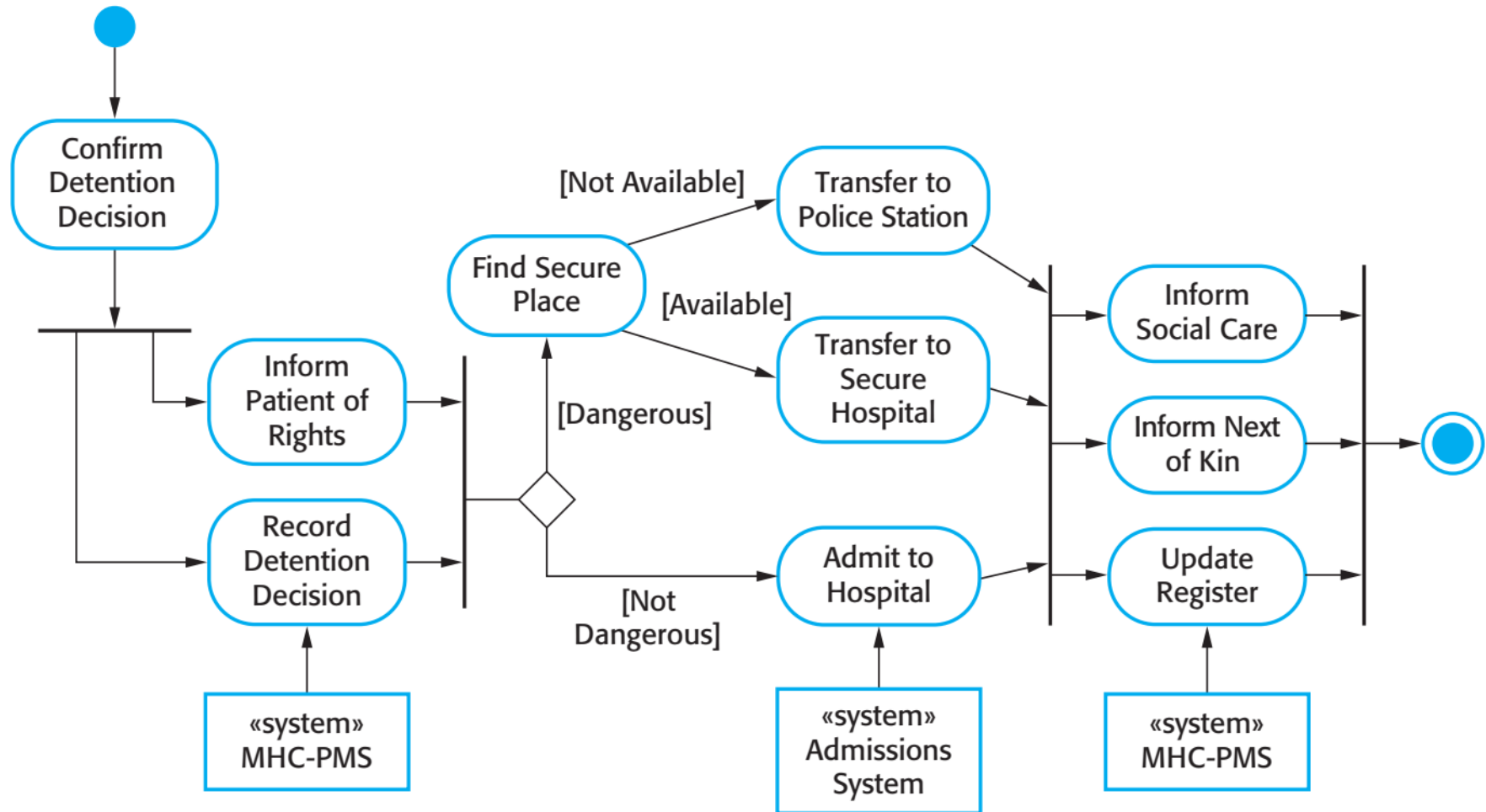
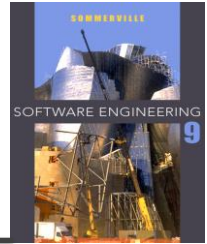
# Süreç Perspektifi

---



- Bağlam modelleri, geliştirilmekte olan sistemin o ortamda nasıl kullanıldığını değil, ortamdaki diğer sistemleri gösterir.
- Süreç modelleri, geliştirilmekte olan sistemin daha geniş iş süreçlerinde nasıl kullanıldığını ortaya çıkarır.
- UML aktivite diyagramları, iş süreci modellerini tanımlamak için kullanılabilir.

# AS-HYS'deki İstemsiz Gözaltı Süreç Modeli



# Etkileşim Modelleri



- Kullanıcı etkileşimlerini modellemek, kullanıcı gereksinimlerinin belirlenmesine yardımcı olduğu için önemlidir.
- Sistemler arası etkileşimi modellemek, ortaya çıkabilecek iletişim sorunlarını vurgular.
- Bileşen etkileşimini modellemek, önerilen bir sistem yapısının gerekli sistem performansını ve güvenilirliği sağlayıp sağlamayacağını anlamamıza yardımcı olur.
- Kullanım durumu diyagramları ve sıra diyagramları etkileşim modellemesi için kullanılabilir.

# Durum Modelleme Kullanın

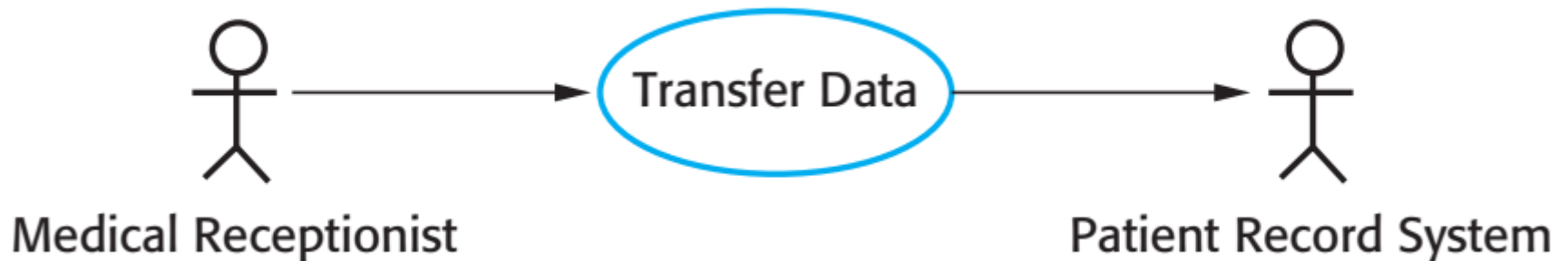


- Kullanım senaryoları, başlangıçta gereksinimlerin ortaya çıkarılmasını desteklemek için geliştirildi ve şimdi UML'ye dahil edildi.
- Her kullanım durumu, bir sistemle harici etkileşimi içeren ayrı bir görevi temsil eder.
- Kullanım durumundaki aktörler, insanlar veya diğer sistemler olabilir.
- Kullanım senaryosu şematik olarak temsil edilir çünkü: genel bir bakış sağlamak için ve daha ayrıntılı bir metin olarak anlatılmak için.

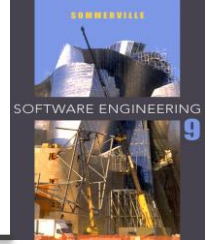
# Veri Aktarımı Kullanım Durumu



- AS-HYS'de bir kullanım durumu



# 'Veri Aktarımı' Kullanım Senaryosunun Tablo Şeklinde Açıklaması

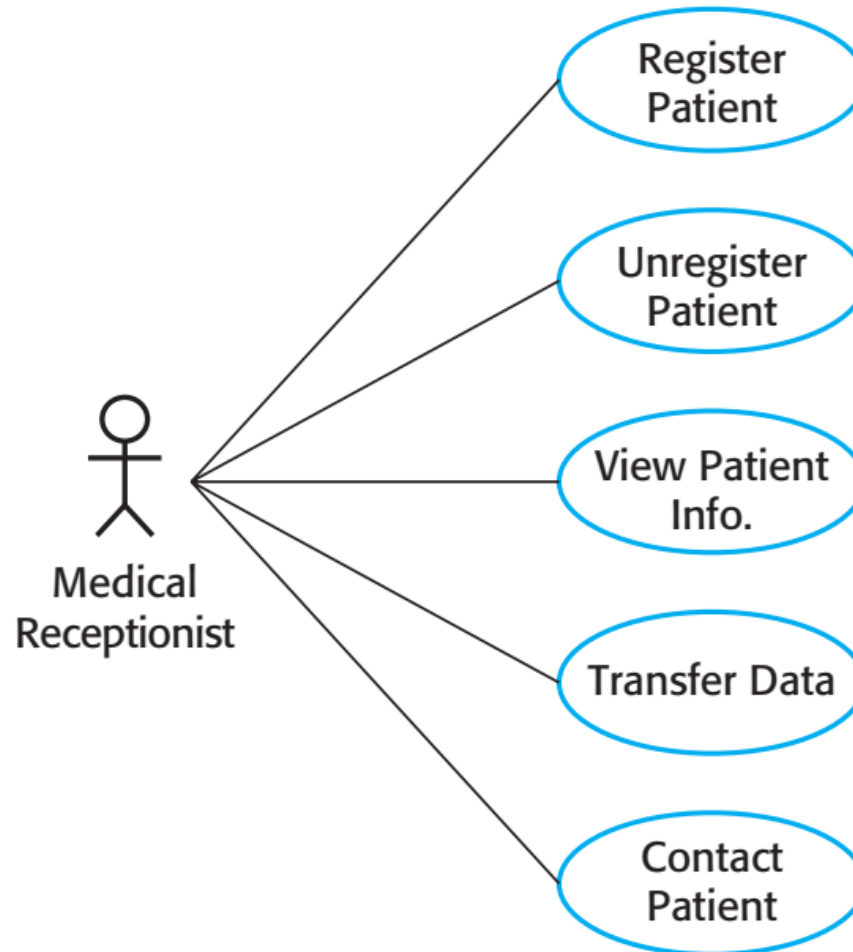


## AS-HYS: Veri Aktarımı

Aktörler	Tıbbi resepsiyon görevlisi, hasta kayıt sistemi (PRS)
Açıklama	Bir resepsiyon görevlisi, AS-HYS'den bir sağlık otoritesi tarafından tutulan genel bir hasta kaydı veri tabanına, veri aktarabilir. Aktarılan bilgiler, güncellenmiş kişisel bilgiler (adres, telefon numarası vb.) veya hastanın tanı ve tedavisinin bir özeti olabilir.
Veri	Hastanın kişisel bilgileri, tedavi özeti
Uyaran	Tıbbi resepsiyon görevlisi tarafından verilen kullanıcı komutu
Tepki	PRS'nin (Hasta Kayıt Sistemi) güncellendiğine dair onay
Yorumlar	Resepsiyon görevlisi, hasta bilgilerine ve PRS'ye erişmek için uygun güvenlik izinlerine sahip olmalıdır.



# AS-HYS'de 'Tıbbi Resepsiyonist' Rolünü İçeren Kullanım Durumları

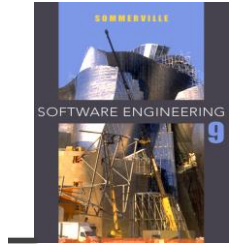


# Sıra Diyagramları

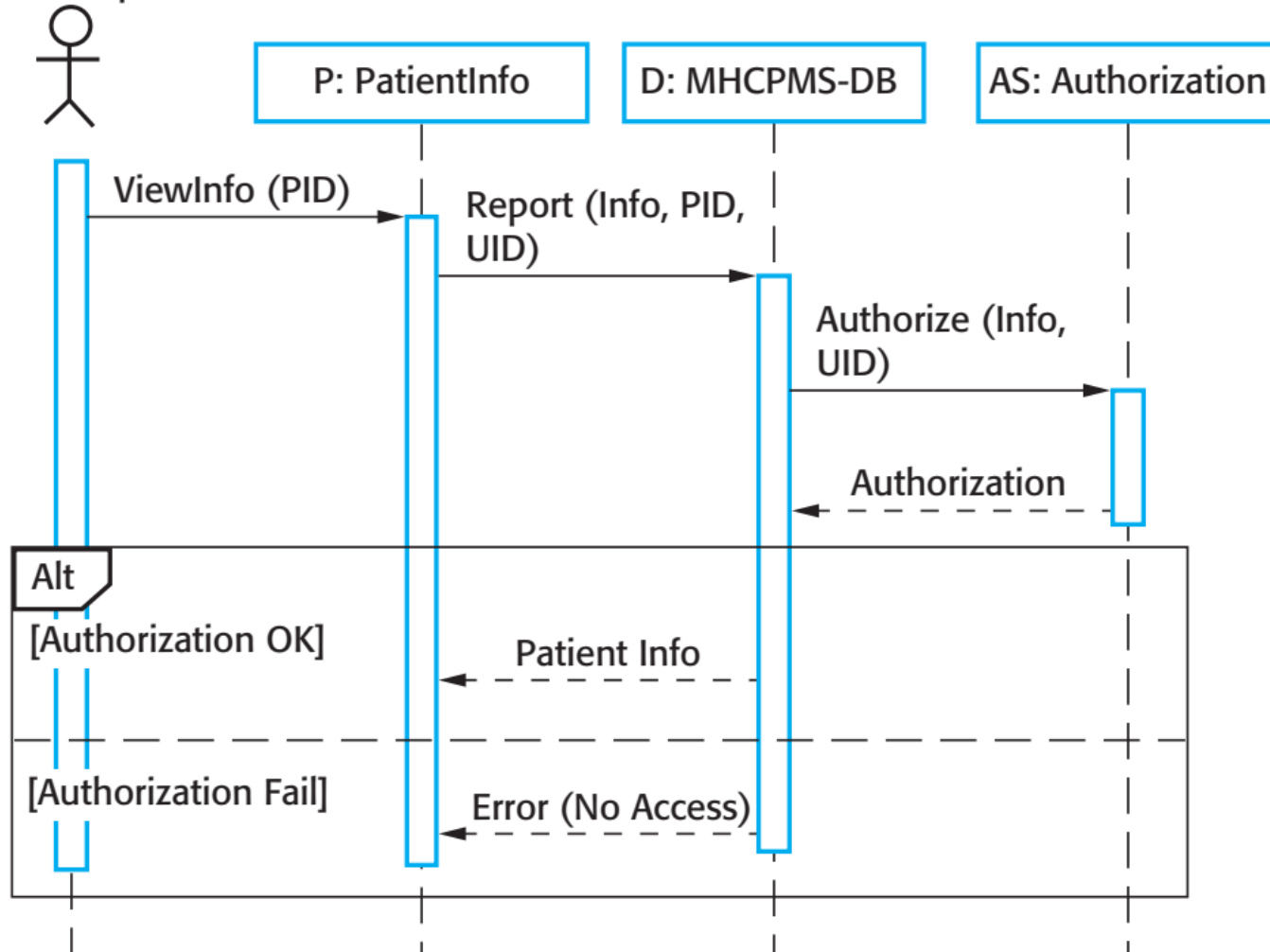


- Sıra diyagramları, UML'nin bir parçasıdır ve bir sistem içindeki aktörler ve nesneler arasındaki etkileşimleri modellemek için kullanılır.
- Bir dizi diyagramı, belirli bir kullanım senaryosu veya kullanım örneği sırasında meydana gelen etkileşimlerin sırasını gösterir.
- İlgili nesneler ve aktörler, bunlardan dikey olarak çizilmiş noktalı bir çizgi ile diyagramın üst kısmında listelenir.
- Nesneler arasındaki etkileşimler, açıklamalı oklarla belirtilmiştir.

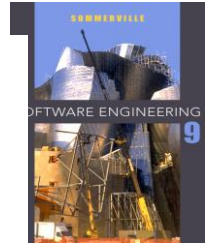
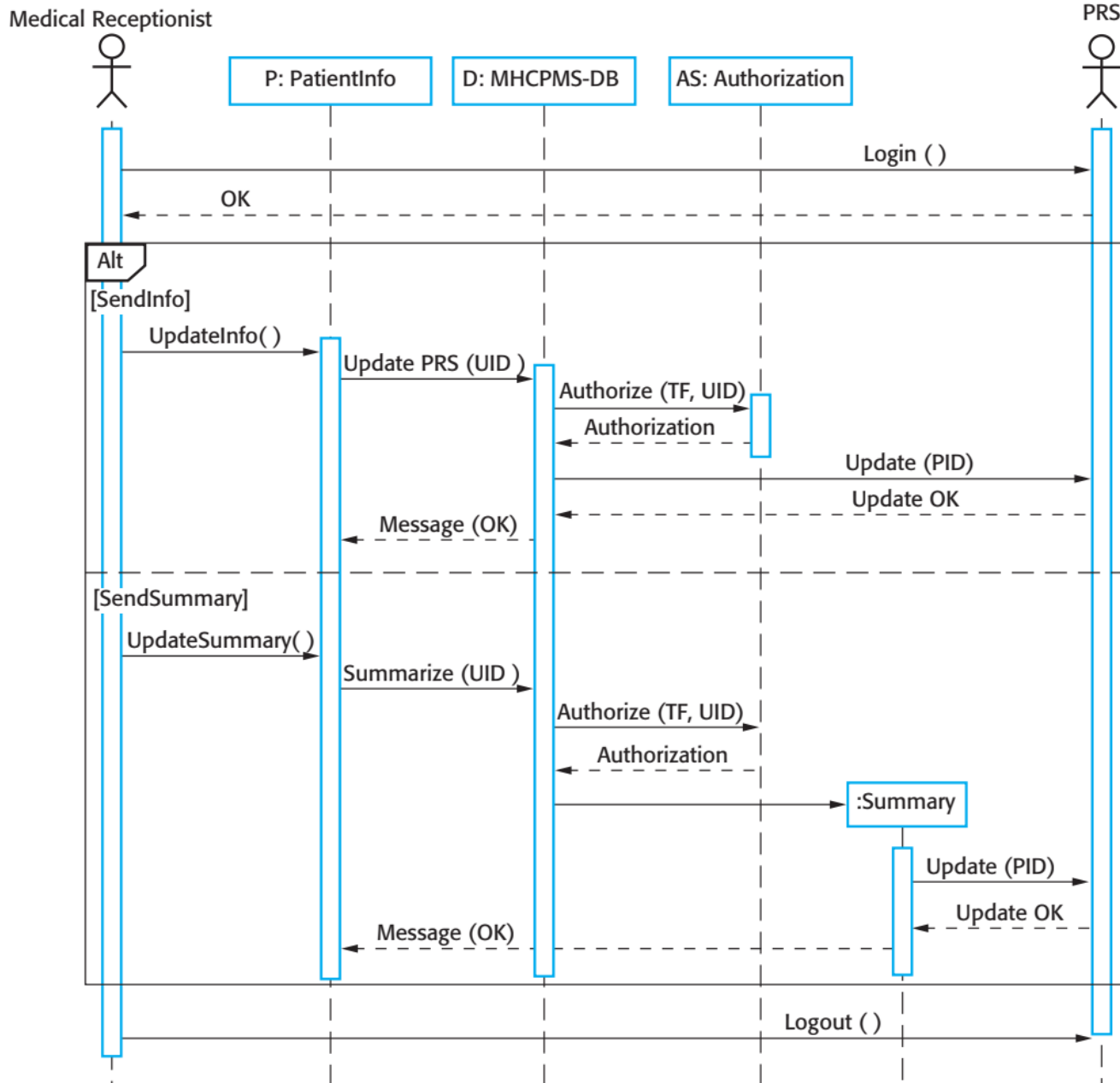
# Hasta Bilgilerini Görüntülemek İçin Sıra Diyagramı



Medical Receptionist



# Aktarım Verileri İçin Sıra Diyagramı



# Yapısal Modeller

---



- Yapısal yazılım modelleri, bir sistemin organizasyonunu, o sistemi oluşturan bileşenler ve bunların ilişkileri açısından gösterir.
- Yapısal modeller, sistem tasarımının yapısını gösteren statik modeller veya sistemin yürütülürken organizasyonunu gösteren dinamik modeller olabilir.
- Sistem mimarisini tartışırken ve tasarlarken bir sistemin yapısal modellerini oluşturursunuz.

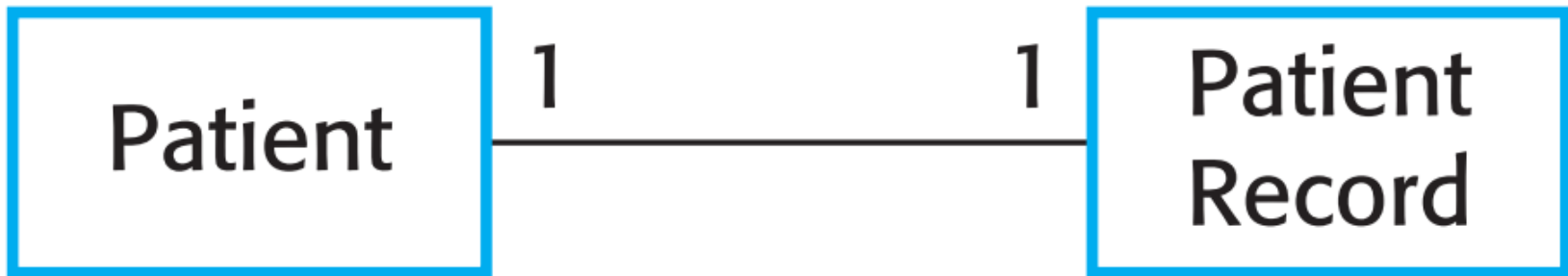
# Sınıf Diyagramları



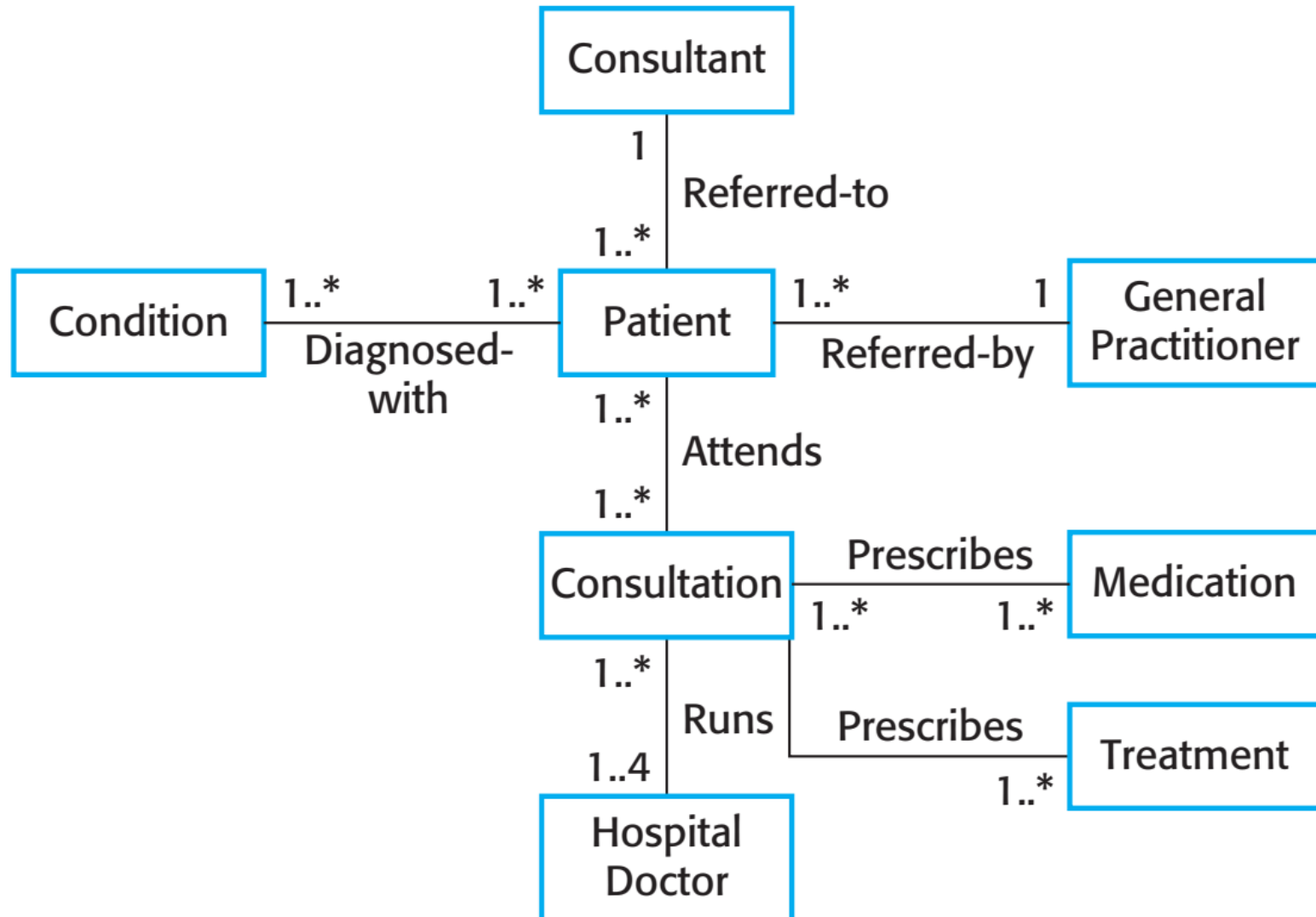
- Sınıf diyagramları, bir sistemdeki sınıfları ve bu sınıflar arasındaki ilişkileri göstermek için nesne yönelimli bir sistem modeli geliştirirken kullanılır.
- Bir nesne sınıfı, bir tür sistem nesnesinin genel bir tanımı olarak düşünülebilir.
- İlişki, bu sınıflar arasında bir ilişki olduğunu gösteren, sınıflar arasındaki bir bağlantıdır.
- Yazılım mühendisliği sürecinin ilk aşamalarında modeller geliştirirken, nesneler hasta, reçete, doktor vb. gibi gerçek dünyadaki bir şeyi temsil eder.

# UML Sınıfları ve İlişkilendirmesi

---



# AS-HYS'deki Sınıflar ve Bağlantıları





# Danışma Sınıfı



## Consultation

Doctors  
Date  
Time  
Clinic  
Reason  
Medication Prescribed  
Treatment Prescribed  
Voice Notes  
Transcript  
...

New ()  
Prescribe ()  
RecordNotes ()  
Transcribe ()  
...

# Bölüm 1'in Anahtar Noktaları



- Model, sistem ayrıntılarını göz ardı eden bir sistemin soyut bir görünümüdür. Sistemin bağlamını, etkileşimlerini, yapısını ve davranışını göstermek için tamamlayıcı sistem modelleri geliştirilebilir.
- Bağlam modelleri, modellenmekte olan bir sistemin diğer sistem ve süreçlerle birlikte bir ortamda nasıl konumlandırıldığını gösterir.
- Kullanım durumu diyagramları ve sıra diyagramları, tasarlanmakta olan sistemdeki kullanıcılar ve sistemler arasındaki etkileşimleri tanımlamak için kullanılır. Kullanım senaryoları, bir sistem ile dış aktörler arasındaki etkileşimleri tanımlar; sıra diyagramları, sistem nesneleri arasındaki etkileşimleri göstererek bunlara daha fazla bilgi ekler.
- Yapısal modeller, bir sistemin organizasyonunu ve mimarisini gösterir. Sınıf diyagramları, bir sistemdeki sınıfların statik yapısını ve ilişkilerini tanımlamak için kullanılır.

---

## Ders 5 - Sistem Modelleme

# Bölüm 2

# Genelleştirme



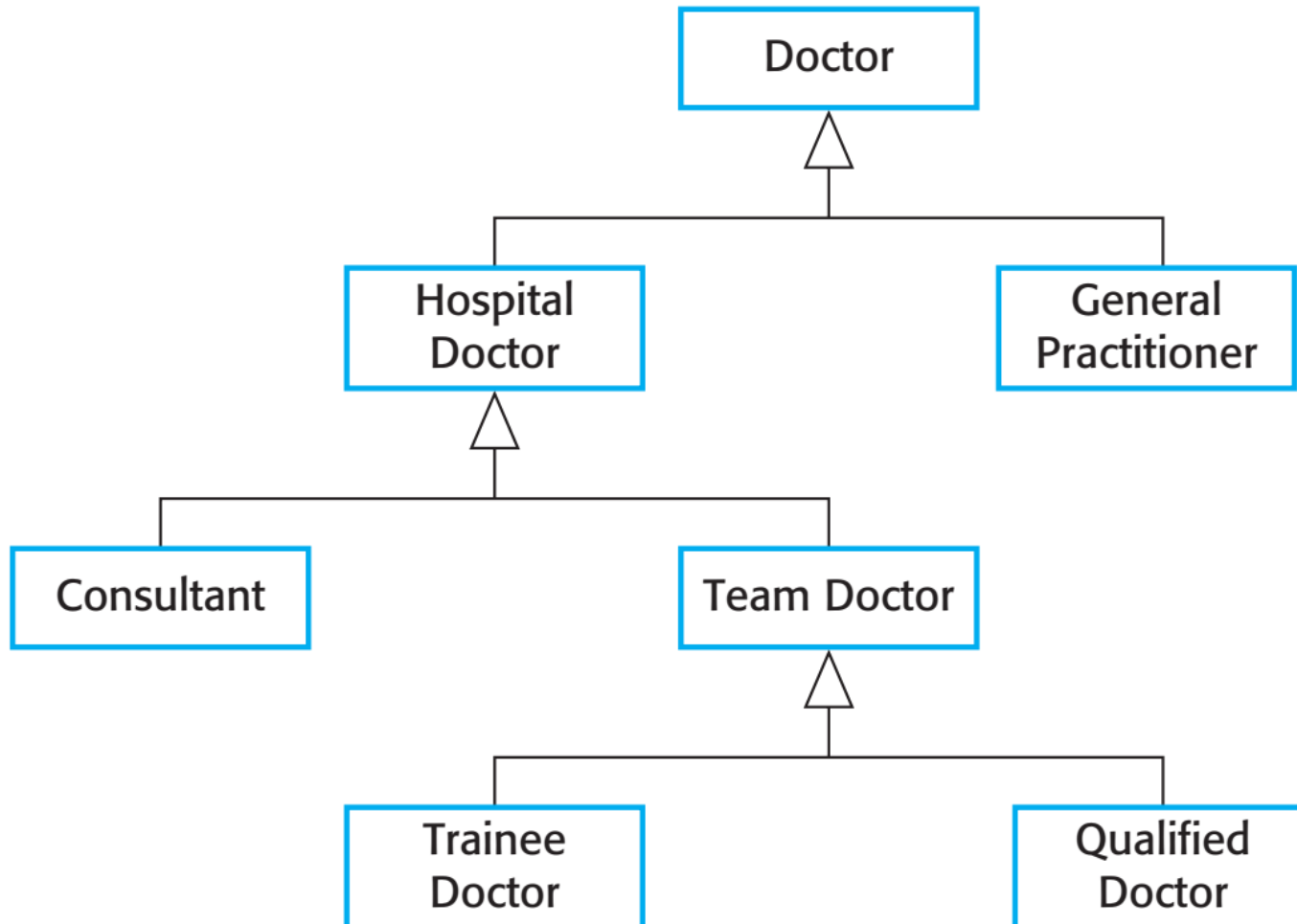
- Genelleştirme, karmaşıklığı yönetmek için kullandığımız günlük bir tekniktir.
- Yaşadığımız her varlığın ayrıntılı özelliklerini öğrenmek yerine, bu varlıkları daha genel sınıflara (hayvanlar, arabalar, evler, vb.) yerleştiriyor ve bu sınıfların özelliklerini öğreniyoruz.
- Bu, bu sınıfların farklı üyelerinin bazı ortak özelliklere sahip olduğu sonucuna varmamızı sağlar, örneğin sincaplar ve sıçanlar kemirgenlerdir.

# Genelleştirme

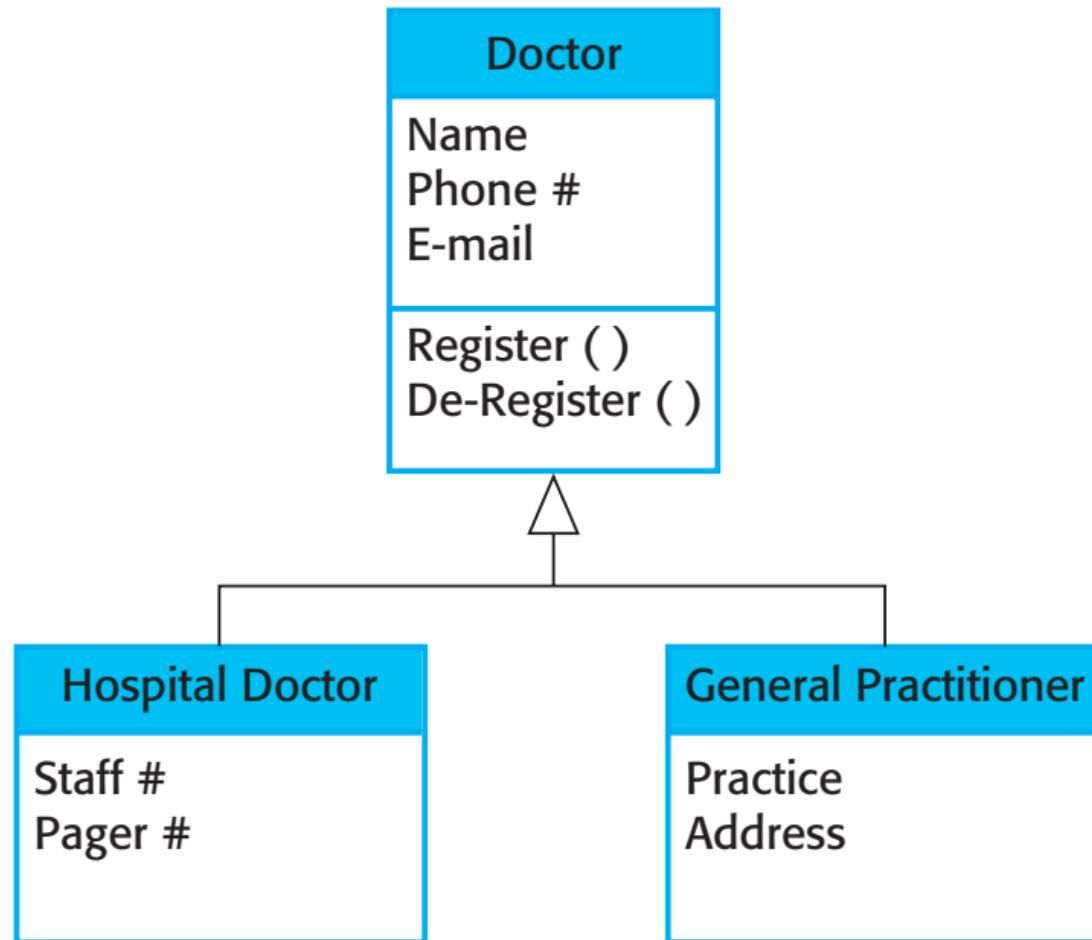
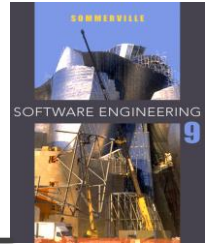


- Modelleme sistemlerinde, genelleştirme kapsamı olup olmadığını görmek için bir sistemdeki sınıfları incelemek genellikle yararlıdır. Değişiklikler önerilirse, değişiklikten etkilenip etkilenmediklerini görmek için sistemdeki tüm sınıflara bakmanıza gerek yoktur.
- Java gibi nesne yönelimli dillerde genelleştirme, dilde yerleşik olan sınıf kalıtım mekanizmaları kullanılarak gerçekleştirilir.
- Bir genelleştirmede, üst düzey sınıflarla ilişkili öznitelikler ve işlemler de alt düzey sınıflarla ilişkilendirilir.
- Alt düzey sınıflar, alt sınıflardır, nitelikleri ve işlemleri üst sınıflarından devralır. Bu alt düzey sınıflar daha sonra daha özel nitelikler ve işlemler ekler.

# Bir Genelleştirme Hiyerarşisi



# Ek Ayrıntı İçeren Bir Genelleştirme Hiyerarşisi



# Nesne Sınıfı Birleştirme Modelleri

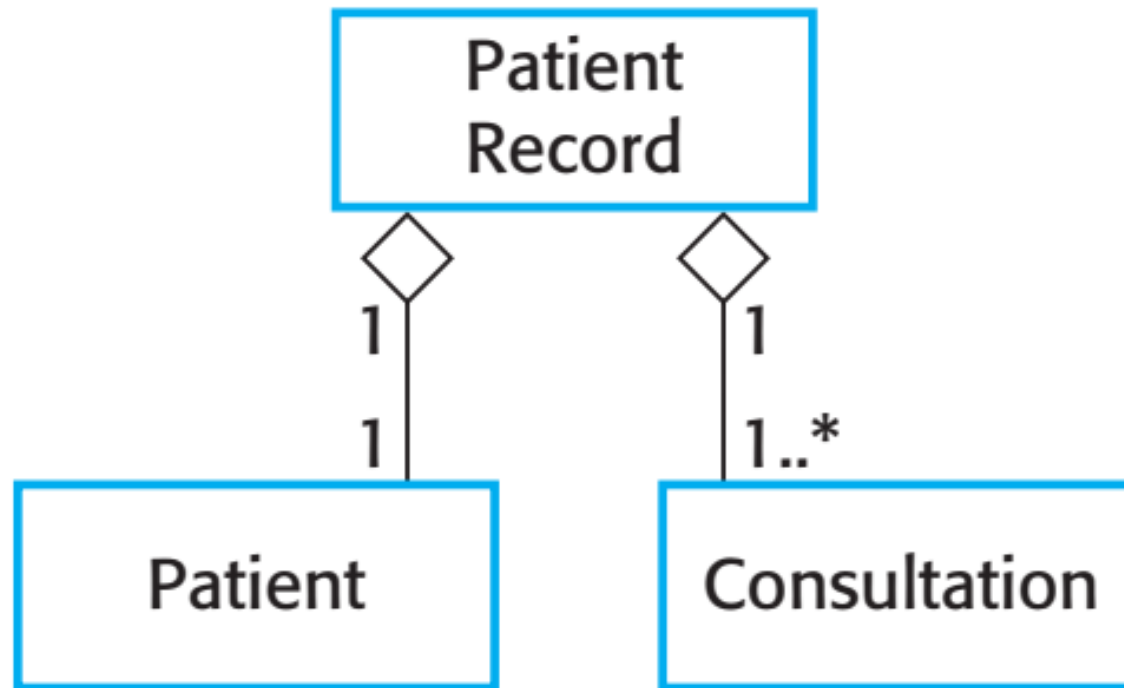
---



- Bir toplama modeli, koleksiyon olan sınıfların diğer sınıflardan nasıl oluştuğunu gösterir.
- Toplama modelleri, anlamsal veri modellerindeki ilişkinin parçası ile benzerdir.



# Veri Toplama İlişkisi



# Davranışsal Modeller



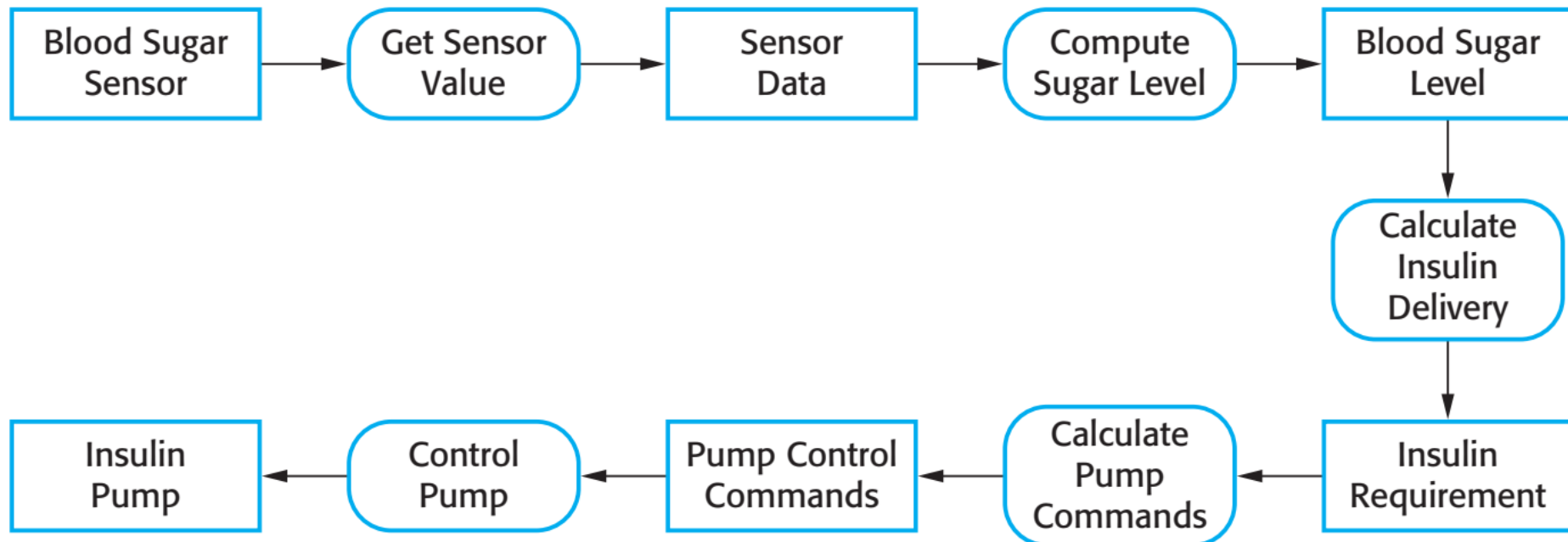
- Davranış modelleri, bir sistemin yürütülürken dinamik davranışının modelleridir. Bir sistem, çevresinden gelen bir uyarana tepki verdiğinde ne olduğunu veya ne olması gerektiğini gösterirler.
- Bu uyarıların iki tür olduğunu düşünebilirsiniz:
  - **Veriler** Sistem tarafından işlenmesi gereken bazı veriler gelir.
  - **Olaylar** Sistem işlemeyi tetikleyen bazı olaylar meydana gelir. Her zaman böyle olmasa da, olayların ilişkili verileri olabilir.

# Veriye Dayalı Modelleme



- Pek çok iş sistemi, öncelikli olarak veriler tarafından yönlendirilen veri işleme sistemleridir. Görece az harici olay işleme ile sisteme veri girişi ile kontrol edilirler.
- Veriye dayalı modeller, girdi verilerinin işlenmesinde ve ilişkili bir çıktının oluşturulmasında yer alan eylemlerin sırasını gösterir.
- Bir sistemde uçtan uca işlemeyi göstermek için kullanılabildiklerinden, gereksinimlerin analizi sırasında özellikle yararlıdırlar.

# İnsülin Pompasının Çalışmasının Bir Aktivite Modeli

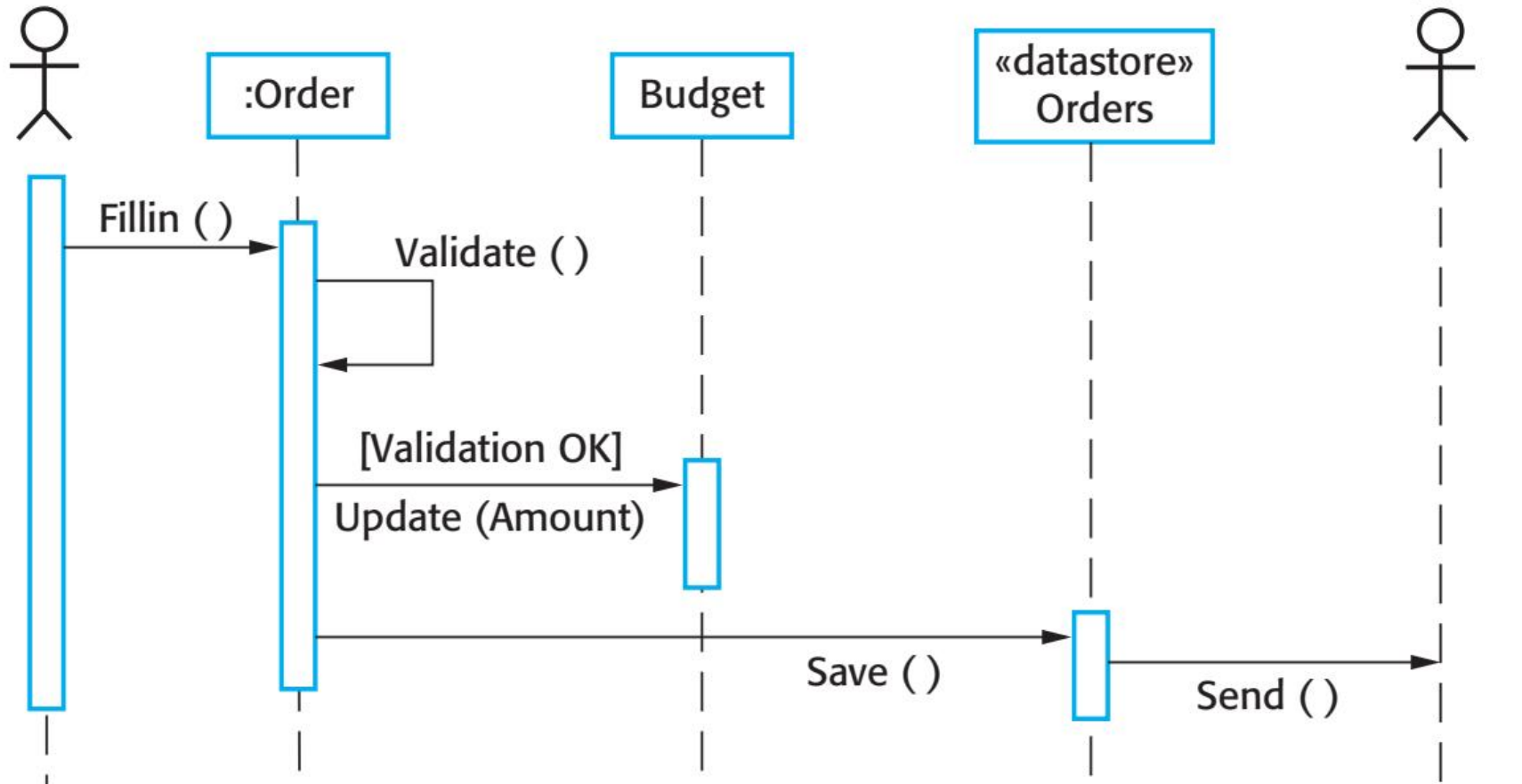


# Sipariş İşleme Süreci



Purchase Officer

Supplier



# Olay Odaklı Modelleme



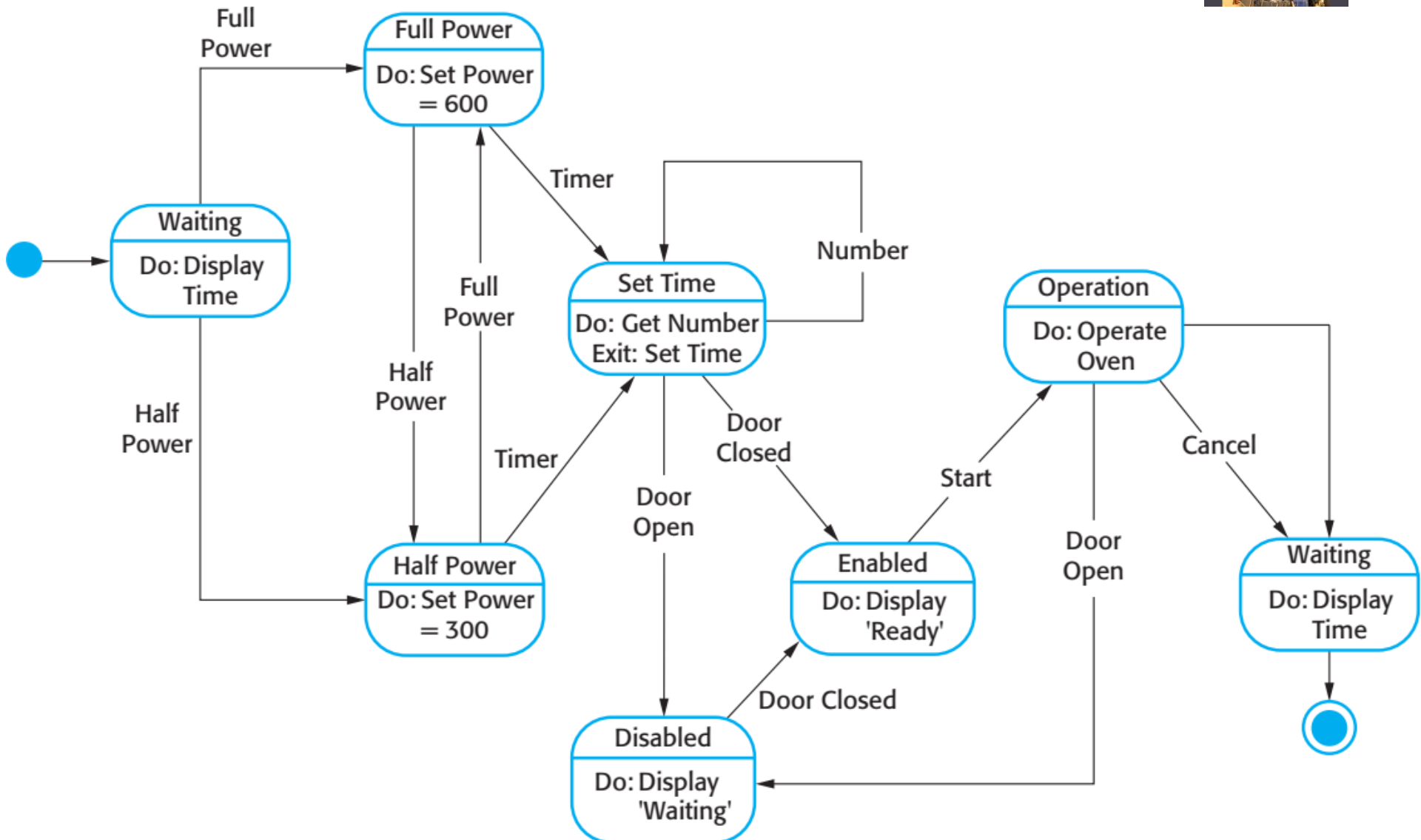
- Gerçek zamanlı sistemler genellikle minimum veri işleme ile olay güdümlüdür. Örneğin, bir sabit hat telefon değiştirme sistemi, bir çevir sesi üreterek "alıcı açık" gibi olaylara yanıt verir.
- Olay güdümlü modelleme, bir sistemin harici ve dahili olaylara nasıl tepki verdiğini gösterir.
- Bir sistemin sınırlı sayıda duruma sahip olduğu ve olayların (uyaranların) bir durumdan diğerine geçişe neden olabileceği varsayımına dayanır.

# Durum Makine Modelleri



- Bunlar, harici ve dahili olaylara yanıt olarak sistemin davranışını modeller.
- Sistemin uyarılara tepkilerini gösterirler, bu nedenle genellikle gerçek zamanlı sistemleri modellemek için kullanılırlar.
- Durum makinesi modelleri, sistem durumlarını düğümler olarak ve olayları bu düğümler arasındaki yaylar olarak gösterir. Bir olay meydana geldiğinde, sistem bir durumdan diğerine geçer.
- Statechart'lar UML'nin ayrılmaz bir parçasıdır ve durum makine modellerini temsil etmek için kullanılır.

# Mikrodalga Fırının Durum Diyagramı





# Mikrodalga Fırın İçin Durumlar ve Uyarıcılar (A)



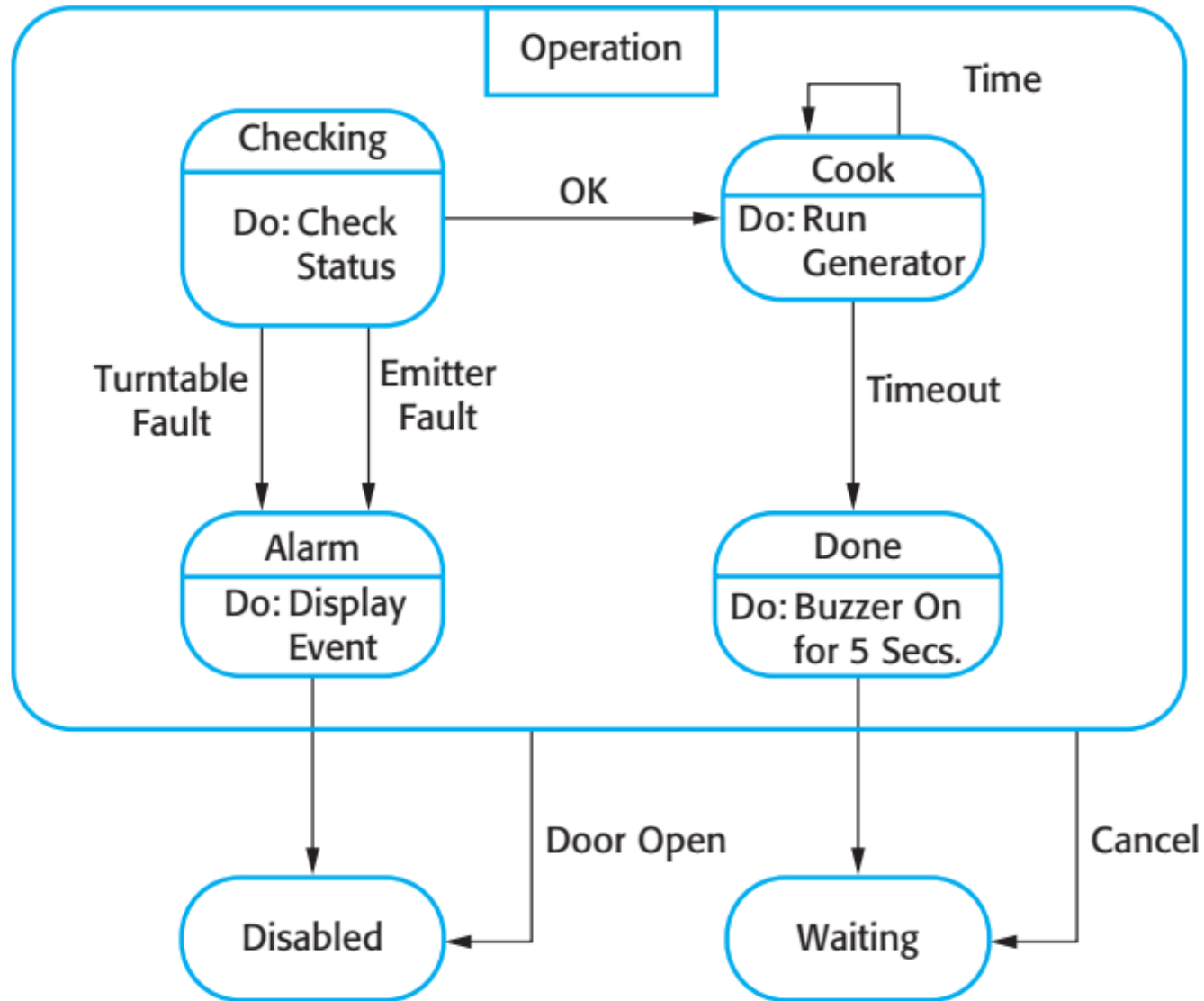
Durum	Açıklama
Bekliyorum	Fırın girdi bekliyor. Ekran güncel saati gösterir.
Yarım güç	Fırın gücü 300 watt olarak ayarlanmıştır. Ekranda 'Yarım güç' gösterilir.
Tam güç	Fırın gücü 600 watt olarak ayarlanmıştır. Ekranda 'Tam güç' gösterilir.
Ayarlanan zaman	Piştirme süresi, kullanıcının girdiği değere ayarlanır. Ekranda seçilen piştirme süresi gösterilir ve saat ayarlandıkça güncellenir.
Devre dışı	Fırın çalışması, güvenlik için devre dışı bırakılmıştır. İç fırın ışığı yanıyor. Ekran 'Hazır değil' gösteriyor.
Etkin	Fırın işletimi etkinleştirildi. İç fırın ışığı sönmük. Ekranda "Piştirmeye hazır" gösterilir.
Operasyon	Fırın çalışıyor. İç fırın ışığı yanıyor. Ekran zamanlayıcı geri sayımını gösterir. Piştirme tamamlandığında, zil beş saniye boyunca çalar. Fırın ışığı yanıyor. Sesli uyarı çalarken ekranda 'Piştirme tamamlandı' mesajı görünüyor.

# Mikrodalga Fırın İçin Durumlar ve Uyarıcılar (B)



Uyaran	Açıklama
Yarım güç	Kullanıcı yarım güç düğmesine bastı.
Tam güç	Kullanıcı tam güç düğmesine bastı.
Zamanlayıcı	Kullanıcı, zamanlayıcı düğmelerinden birine bastı.
Numara	Kullanıcı bir sayısal tuşa bastı.
Kapı açık	Fırın kapısı anahtarı kapatılmamış.
Kapı kapalı	Fırın kapısı anahtarı kapalıdır.
Başlat	Kullanıcı Başlat düğmesine bastı.
İptal etmek	Kullanıcı İptal düğmesine bastı.

# Mikrodalga Fırının Çalışması



# Model GÜdümlü Mühendislik



- Model güdümlü mühendislik (MGM), geliştirme sürecinin temel çıktılarının programlardan çok modellerin olduğu bir yazılım geliştirme yaklaşımıdır.
- Bir donanım / yazılım platformunda çalışan programlar daha sonra modellerden otomatik olarak oluşturulur.
- MGM'nin savunucuları, bunun yazılım mühendisliğinde soyutlama düzeyini yükselttiğini, böylece mühendislerin artık programlama dili ayrıntılarıyla veya yürütme platformlarının özellikleriyle ilgilenmelerine gerek kalmadığını savunuyorlar.

# Model Güdümlü Mühendisliğin Kullanımı



- Model güdümlü mühendislik, henüz geliştirmenin erken bir aşamasındadır ve yazılım mühendisliği uygulamaları üzerinde önemli bir etkisinin olup olmayacağı belirsizdir.
- Artıları
  - Sistemlerin daha yüksek soyutlama seviyelerinde değerlendirilmesine izin verir
  - Kodun otomatik olarak oluşturulması, sistemleri yeni platformlara uyarlamanın daha ucuz olduğu anlamına gelir.
- Eksileri
  - Soyutlama için modeller ve uygulama için zorunlu olarak doğru değil.
  - Yeni platformlar için çevirmen geliştirme maliyetleri, kod üretmekten elde edilecek tasarruflardan daha ağır basabilir.

# Model Odaklı Mimari



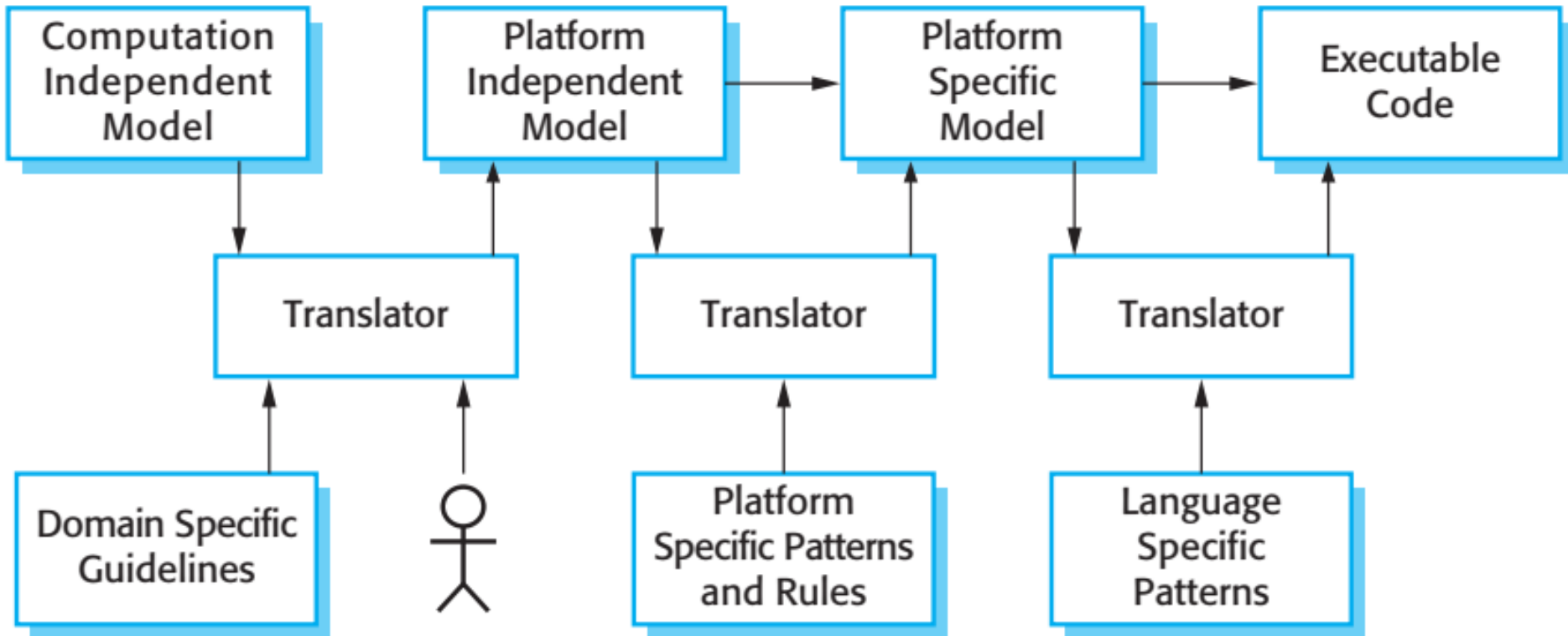
- Model güdümlü mimari (MDA), daha genel model güdümlü mühendisliğin öncüsüydü
- MDA, bir sistemi tanımlamak için UML modellerinin bir alt kümesini kullanan yazılım tasarımı ve uygulamasına yönelik model odaklı bir yaklaşımdır.
- Farklı soyutlama seviyelerinde modeller yaratılır. Yüksek seviyeli, platformdan bağımsız bir modelden, prensip olarak, manuel müdahale olmaksızın bir çalışma programı oluşturmak mümkündür.

# Model Türleri



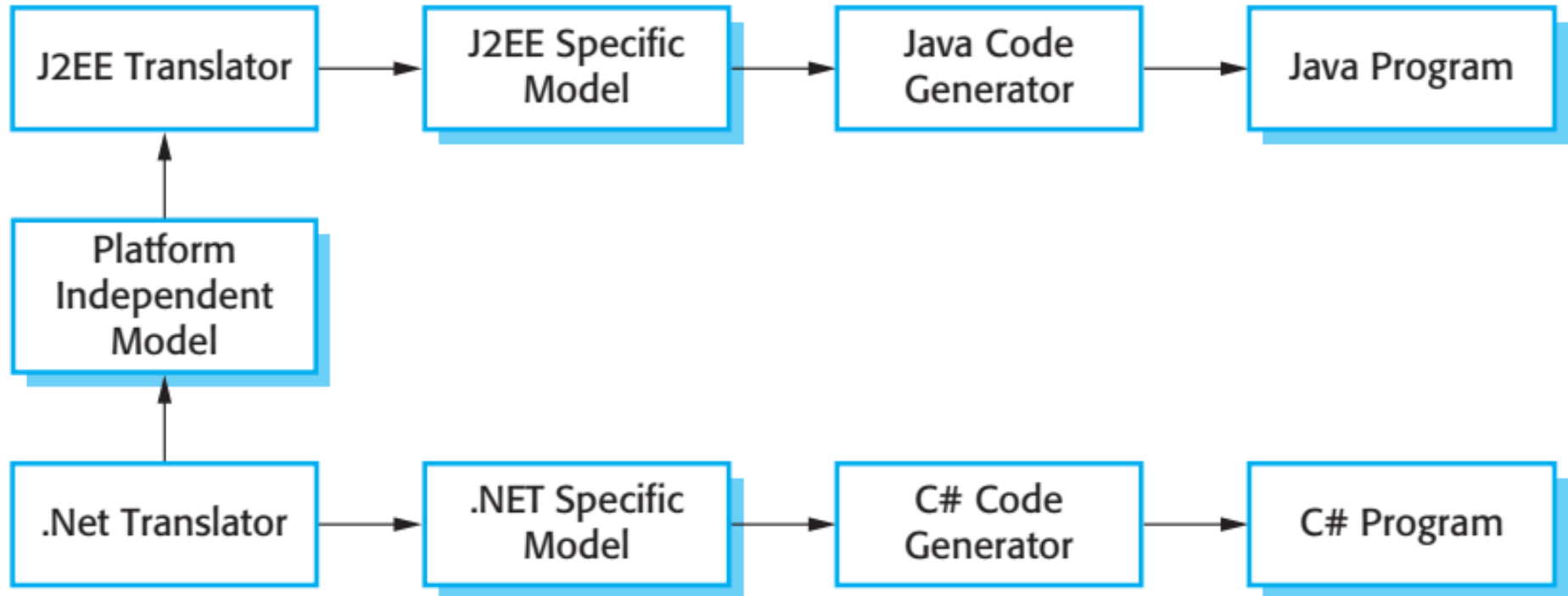
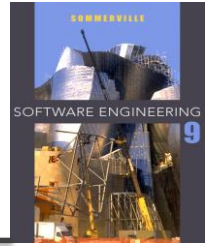
- Hesaplamadan bağımsız bir model (CIM)
  - Bunlar, bir sistemde kullanılan önemli alan soyutlamalarını modeller. CIM'lere bazen etki alanı modelleri denir.
- Platformdan bağımsız bir model (PIM)
  - Bunlar, uygulanmasına referans olmadan sistemin işleyişini modellemektedir. PIM genellikle statik sistem yapısını ve harici ve dahili olaylara nasıl tepki verdiğini gösteren UML modelleri kullanılarak tanımlanır.
- *Platforma özel modeller (PSM)*
  - Bunlar, platformdan bağımsız modelin her uygulama platformu için ayrı bir PSM ile dönüşümleridir. Prensip olarak, her bir katmanın platforma özgü bazı ayrıntılar eklediği PSM katmanları olabilir.

# Model GÜdümlü Mimari Dönüşümleri





# Birden Çok Platforma Özgü Modeller



# Çevik Yöntemler ve MDA



- MDA'nın geliştiricileri, geliştirmeye yinelemeli bir yaklaşımı desteklemeyi amaçladığını ve bu nedenle çevik yöntemlerde kullanılabileceğini iddia ediyor.
- Kapsamlı önden modelleme kavramı, çevik manifestosundaki temel fikirlerle çelişiyor ve çok az çevik geliştiricinin model güdümlü mühendislik konusunda rahat hissettiğinden şüpheleniyorum.
- Dönüşümler tamamen otomatikleştirilebiliyorsa ve bir PIM (platform bağımsız model)'den tam bir program oluşturulabiliyorsa, o zaman prensip olarak MDA, ayrı bir kodlama gerekmeyeceğinden çevik bir geliştirme sürecinde kullanılabilir.

# Yürütülebilir UML

---



- Model güdümlü mühendisliğin arkasındaki temel fikir, modellerin koda tamamen otomatik olarak dönüştürülmesinin mümkün olmasıdır.
- Bu, Yürütülebilir UML veya xUML adı verilen bir UML 2 alt kümesi kullanılarak mümkündür.

# Yürütülebilir UML'nin Özellikleri



- Yürütülebilir bir UML alt kümesi oluşturmak için, model türlerinin sayısı bu nedenle önemli ölçüde şu 3 anahtar türüne indirgenmiştir:
  - Bir sistemdeki temel endişeleri tanımlayan alan modelleri. UML sınıf diyagramları kullanılarak tanımlanırlar ve nesneleri, öznitelikleri ve ilişkileri içerirler.
  - Nitelikleri ve işlemleriyle birlikte sınıfların tanımlandığı sınıf modelleri.
  - Bir durum diyagramının her sınıfla ilişkilendirildiği ve sınıfın yaşam döngüsünü tanımlamak için kullanıldığı durum modelleri.
- Sistemin dinamik davranışı, nesne kısıtlama dili (OCL) kullanılarak bildirimsel olarak belirtilebilir veya UML'nin eylem dili kullanılarak ifade edilebilir.

# Bölüm 2'in Anahtar Noktaları



- Davranış modelleri, bir yürütme sisteminin dinamik davranışını tanımlamak için kullanılır. Bu davranış, sistem tarafından işlenen verilerin perspektifinden veya bir sistemden yanıtları harekete geçiren olaylarla modellenenebilir.
- Etkinlik diyagramları, her bir faaliyetin bir işlem adımını temsil ettiği verilerin işlenmesini modellemek için kullanılabilir.
- Durum diyagramları, bir sistemin iç veya dış olaylara yanıt olarak davranışını modellemek için kullanılır.
- Model odaklı mühendislik, bir sistemin otomatik olarak yürütülebilir koda dönüştürülebilen bir dizi model olarak temsil edildiği yazılım geliştirme yaklaşımıdır.