Bölüm 5

Sistem Modelleme (System Modeling)

Hedefler

- Grafiksel modellerin yazılım sistemlerini tanımlamada nasıl kullanıldığını anlamak
- Farklı tipte modellere neden gerek duyulduğunu ve içerik, etkileşim, yapı ve davranış açısından temel sistem modellemeyi anlamak
- UML(Unified Modeling Language)'de bulunan bazı diyagram türlerini tanımak ve sistem modellemede nasıl kullanılacaklarını anlamak
- Sistemin yapısal ve davranışsal modellerden otomatik olarak üretildiği model-tabanlı mühendisliğin altında yatan fikirlerin farkında olmak

İçerik

- İçerik modelleri
- Etkileşim modelleri
- Yapısal modeller
- Davranışsal modeller
- Model-tabanlı mühendislik

Sistem Modelleme

- Sistem modelleme bir sistemin herbiri sisteme farklı perspektiflerden bakan soyut modellerinin geliştirilme sürecidir
- Sistem modelleme genellikle UML'in sunduğu notasyonları temel alarak sistemin grafiksel bir tanımını ortaya koyma işidir
- Fakat bir sistemi matematiksel olarak da modellemek mümkündür, daha detaylı bir sistem tanımı gerekir

Sistem Modelleme

- Modeller gereksinim mühendisliği sürecinde gereksinimlerin türetilmesi için, tasarım sürecinde sistemi geliştirecek mühendislere sistemi tanımlamak için ve gerçekleştirim sonrasında sistemin yapısını ve yaptığı işleri dokümana dönüştürmek için kullanılır
- Hem mevcut hem de geliştirilecek olan sistem için model oluşturulabilir:
  - Mevcut sistem modelleri gereksinim mühendisliği sürecinde kullanılır. Mevcut sistemin eksiklerini ve güçlü yanlarını belirlemeye ve yeni sistemin gereksinimlerini çıkarmaya yarar.
  - Yeni sistem modelleri projenin paydaşlarına gereksinimleri açıklayabilmek için gereksinim mühendisliği sürecinde kullanılır. Mühendisler bu modelleri tasarım teklifi ve geliştirim için sistem dokümantasyonu yapmak için kullanır.
- Model-tabanlı mühendislik sürecinde sistem modelleri, sistemin tamamını veya bir kısmını geliştirmek için kullanılabilir.

Sistem Modelleme

- Bir sistem modelinin en önemli özelliği detayları dışarıda bırakmasıdır
- Sisteme farklı açılardan bakan farklı modeller geliştirebiliriz:
  - Harici perspektif – Sistemin çevresini modelleyebiliriz
  - Etkileşim perspektifi – Sistem ve çevresi veya bileşenleri arasındaki etkileşimi modelleyebiliriz
  - Yapısal perspektif – Sistemin yapısını veya sistemin işlediği verinin yapısını modelleyebiliriz
  - Davranışsal perspektif – Sistemin dinamik davranışını ve olaylara nasıl tepki verdiğini modelleyebiliriz

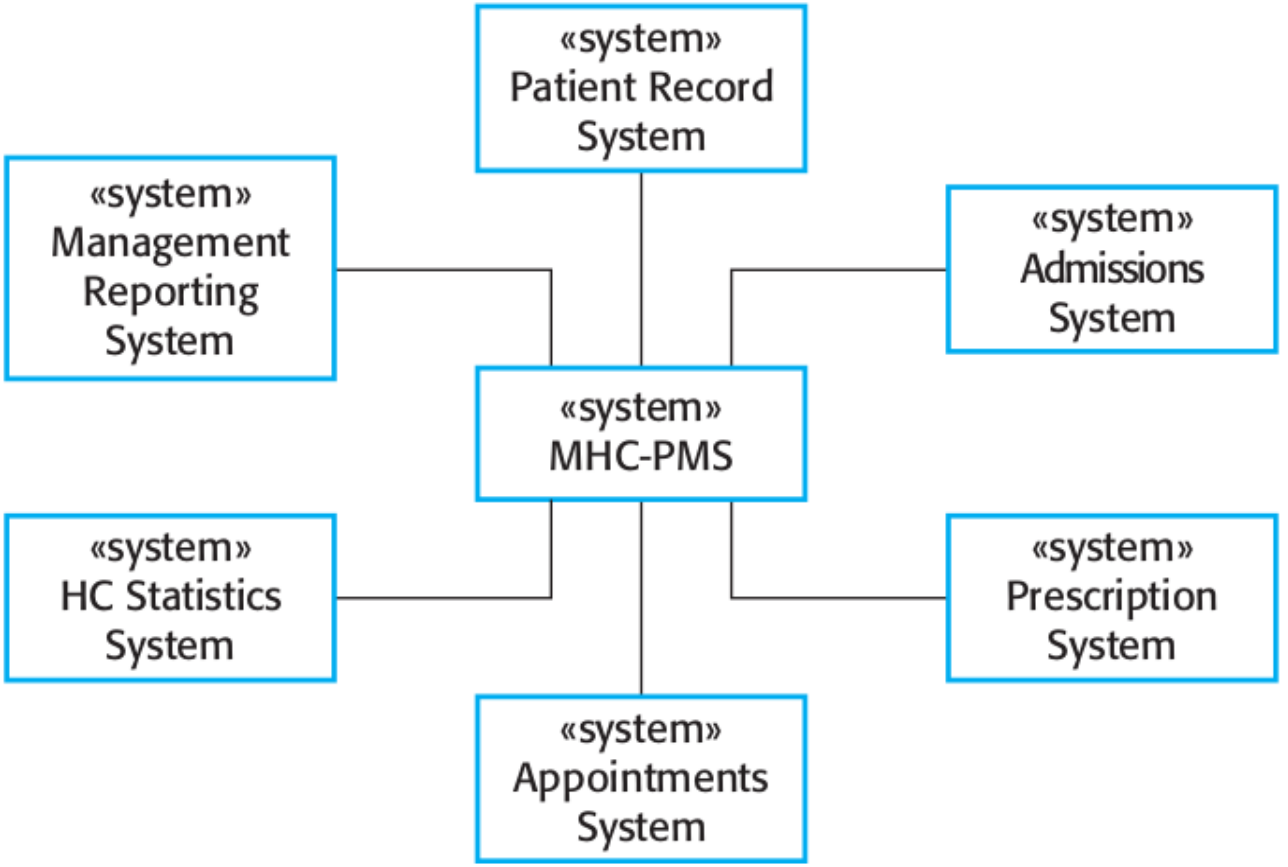
UML (Unified Modelling Language)

- Nesneye dayalı modelleme için kullanılan bir modelleme dili - bir standart haline gelmiştir
- Çoğu UML kullanıcısı, bir sistemin temellerini modelleyebilmek için beş diyagram tipinin yeterli olduğunu belirtmiştir
  - Aktivite (Activity) diyagramları
  - Kullanım durumu (Use case) diyagramları
  - Sıralama (Sequence) diyagramları
  - Sınıf (Class) diyagramları
  - Durum (State) diyagramları

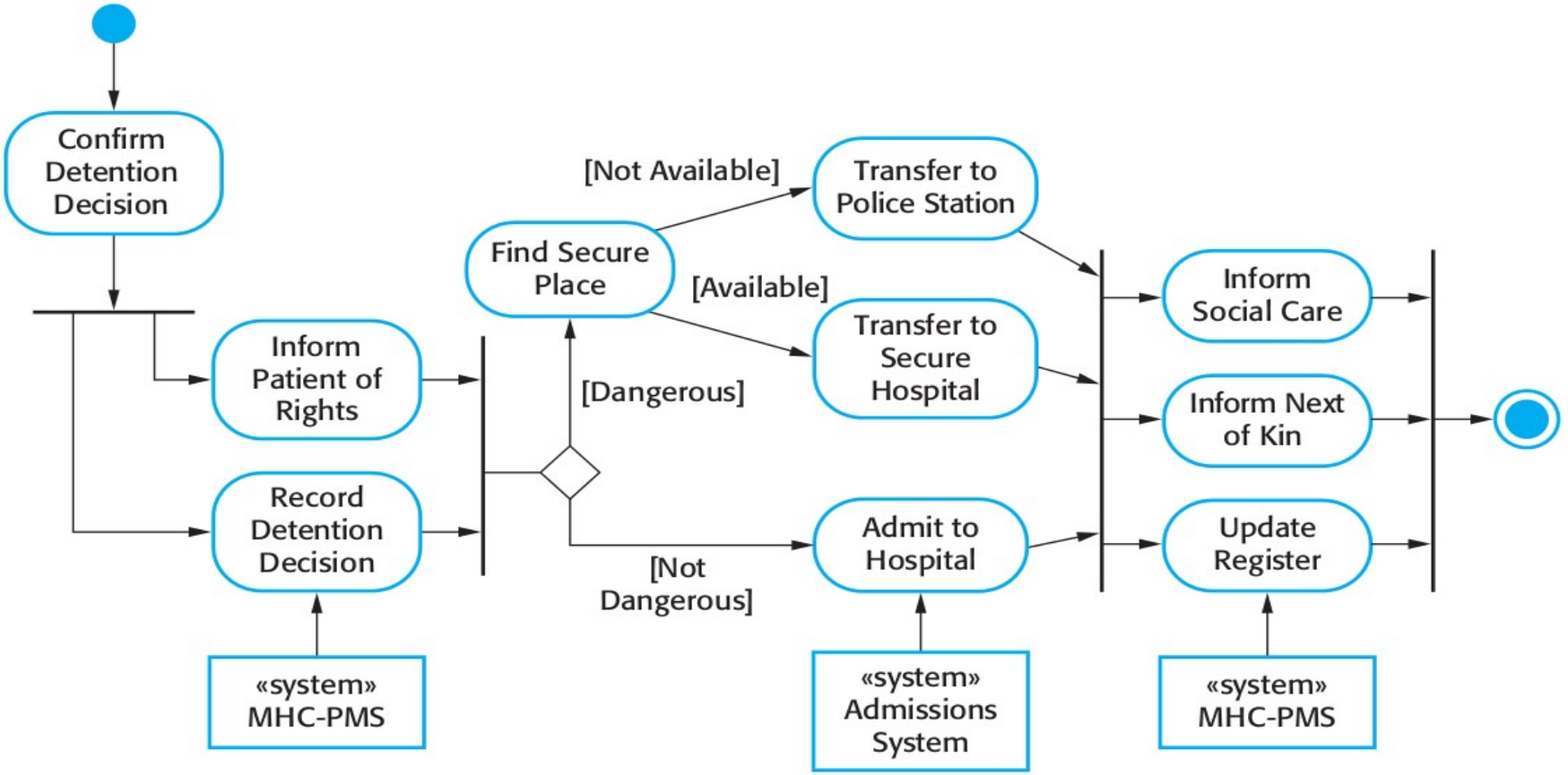
Sistem Modelleme

- Sistem modelleme yaparken grafiksel notasyona bağlı kalma konusunda çok katı olunmayabilir. Modelin ne amaçla kullanılacağı da ne kadar detaylı ve kesin olacağını etkiler. Grafiksel modeller genel olarak üç amaç için kullanılır.
  - Mevcut veya yeni sistem üzerinde tartışabilmek
  - Mevcut sistemin dokümantasyonunu yapmak
  - Sistem gerçekleştiriminde kullanmak

İçerik Modeli



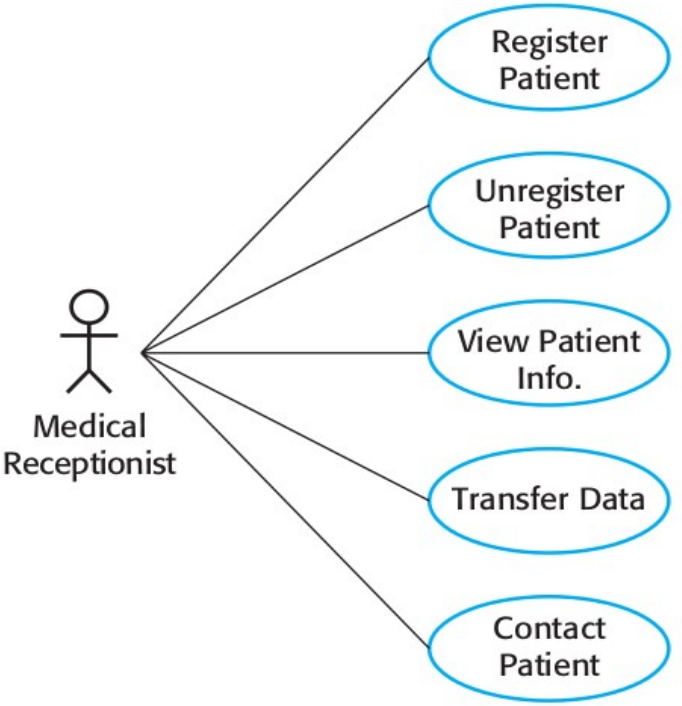
Süreç Modeli



Etkileşim Modelleri

- Kullanım durumu – sistem ve harici aktörler (kullanıcılar veya diğer sistemler) arasındaki etkileşimi göstermek için kullanılır
- Sıralama – sistem bileşenleri arasındaki etkileşimi göstermek için kullanılır (bazen harici bileşenler de yer alabilir)

Kullanım Durumu Diyagramları



Sıralama Diyagramları - 1

1. Tıbbi resepsiyonist, PID (hasta id) ile, PatientInfo sınıfının bir instance'ı olan P üzerinde ViewInfo metodunu çalıştırır

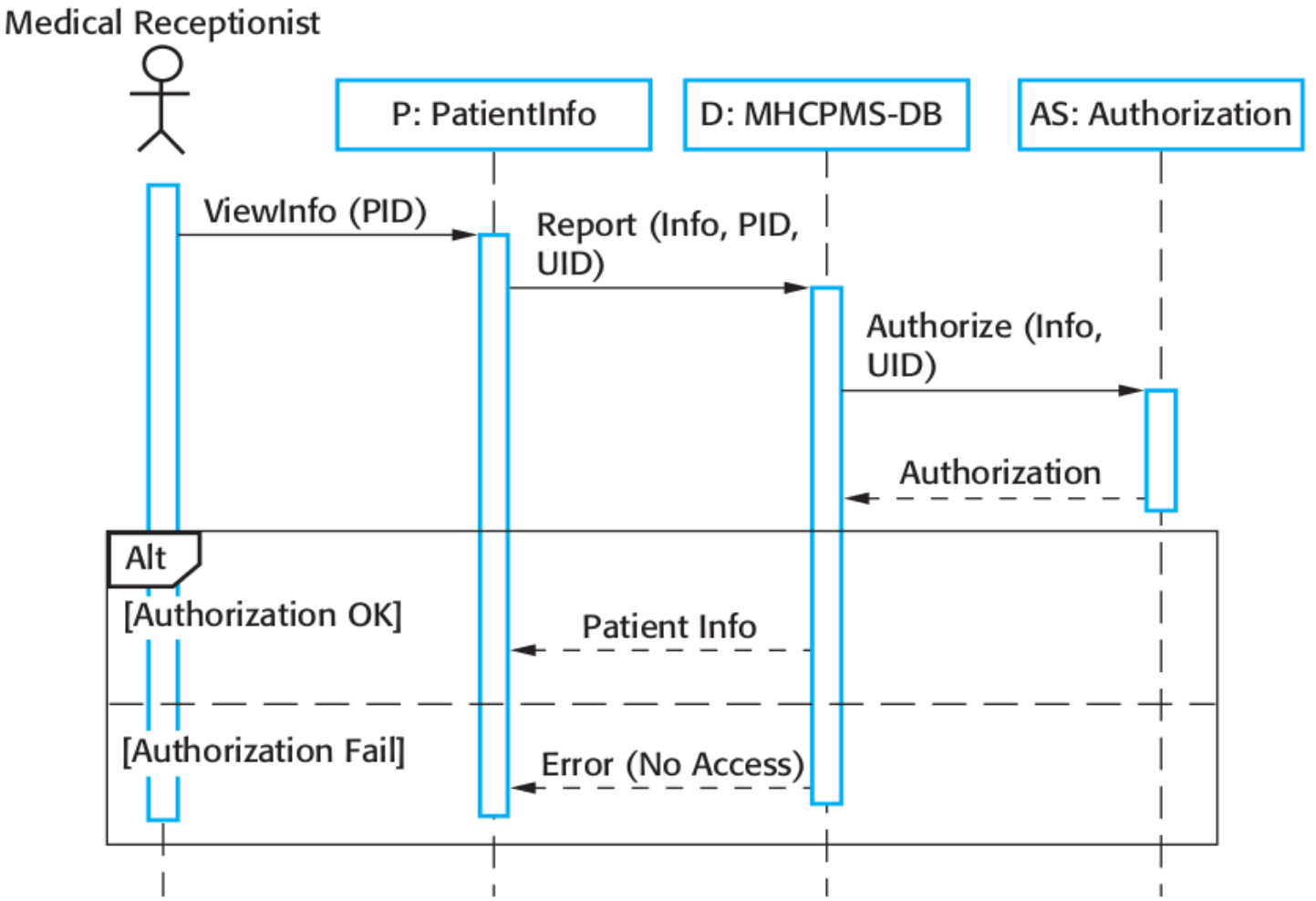
2. P, gerekli bilgileri almak için veritabanını çağırır ve UID (kullanıcı id) bilgisini de güvenlik kontrolü için gönderir

3. Veritabanı kullanıcının bu işlemi yapmaya yetkisinin olup olmadığının kontrolünü yetki sistemi ile yapar.

4. Eğer yetki varsa, hasta bilgileri döndürülür ve kullanıcı ekranındaki form, hasta bilgileri ile doldurulur. Eğer yetki yoksa hata mesajı döndürülür

Sıralama Diyagramları - 1

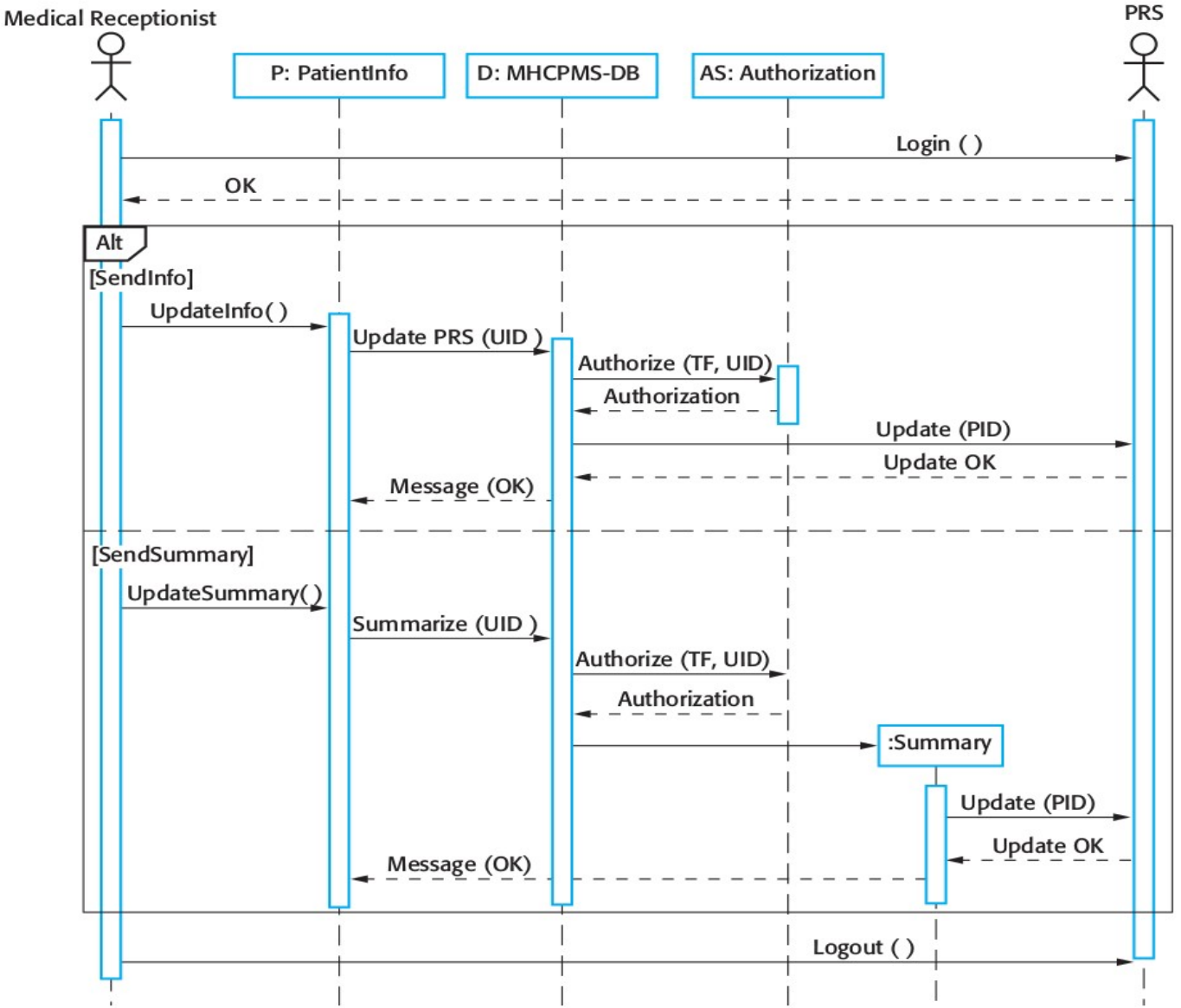
Hasta bilgisi görüntüleme



Sıralama Diyagramları - 2

1. Resepsiyonist sisteme giriş yapar
2. İki seçenek vardır: Birincisi, hasta kayıt sistemine güncellenmiş hasta bilgisini doğrudan gönderir, diğeri de sağlık durumu özetini hasta takip sisteminden çekerek kayıt sistemine aktarır
3. Herbir durumda resepsiyonistin işlem yapmaya yetkili olup olmadığı yetki sistemi aracılığıyla kontrol edilir
4. Kişisel bilgiler ya kullanıcı arayüzünden doğrudan ya da veritabanından özet kaydı oluşturularak hasta kayıt sistemine gönderilir
5. Veri transferi bitince sistem bir durum mesajı gönderir ve kullanıcı sistemden çıkar

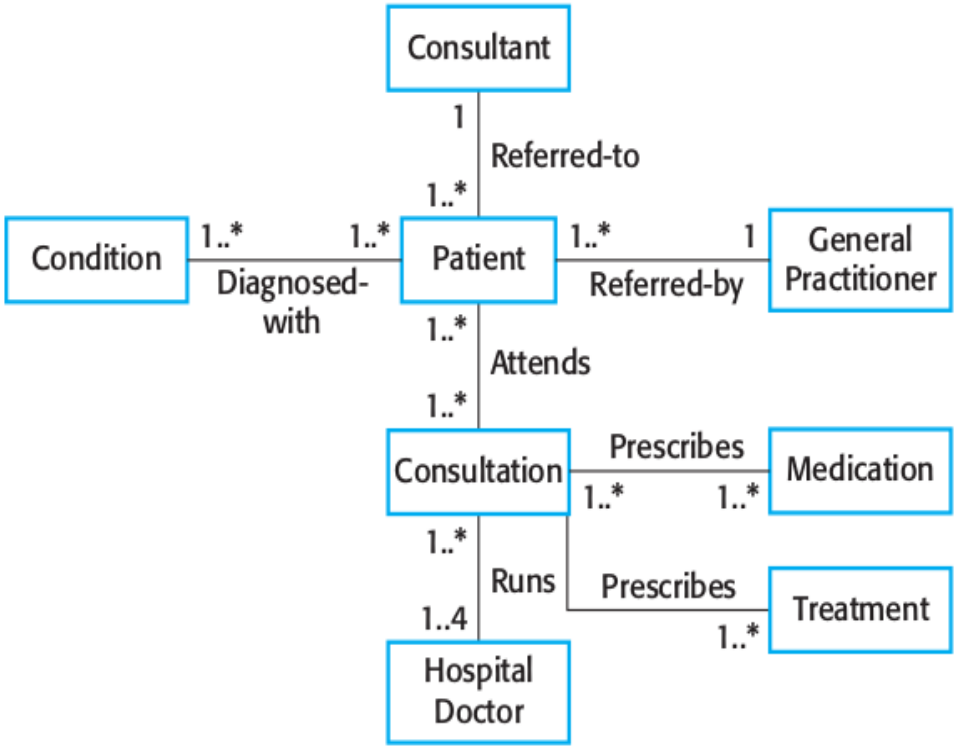
Sıralama Diyagramları - 2

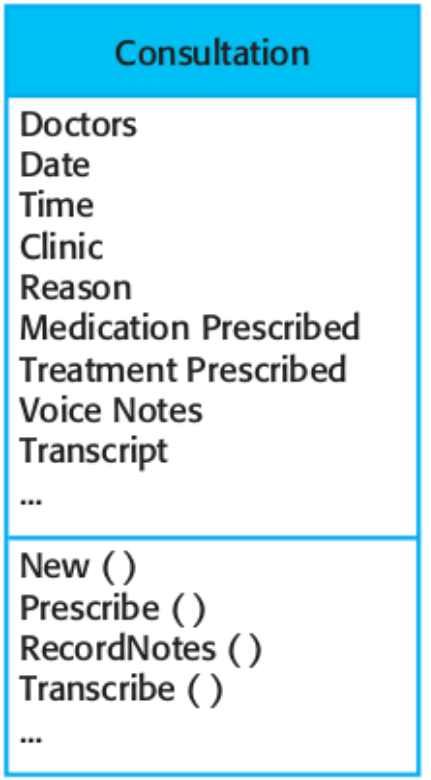


Yapısal Modeller

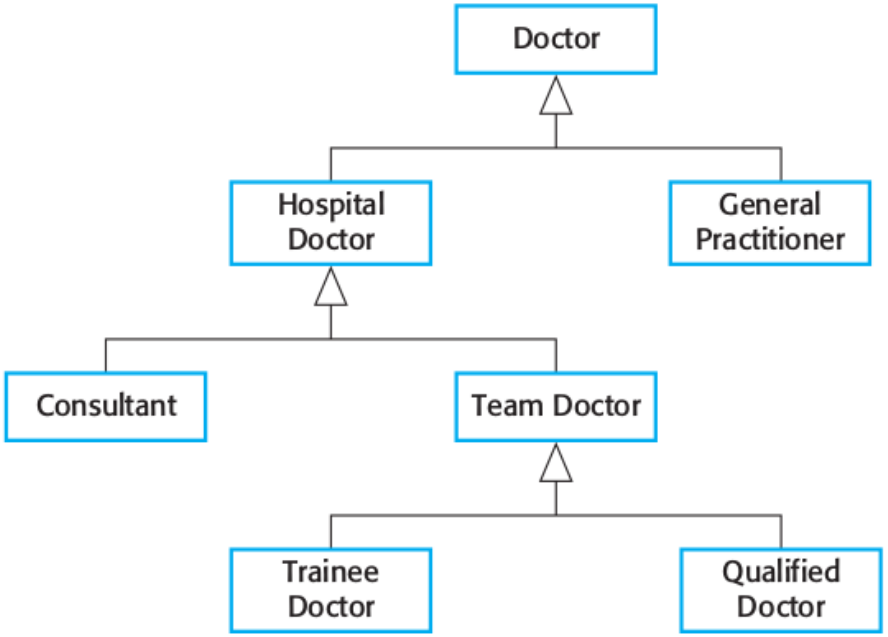
- Sınıf diyagramları
- Genelleme (Generalization)
- Bütünleştirme (Aggregation)

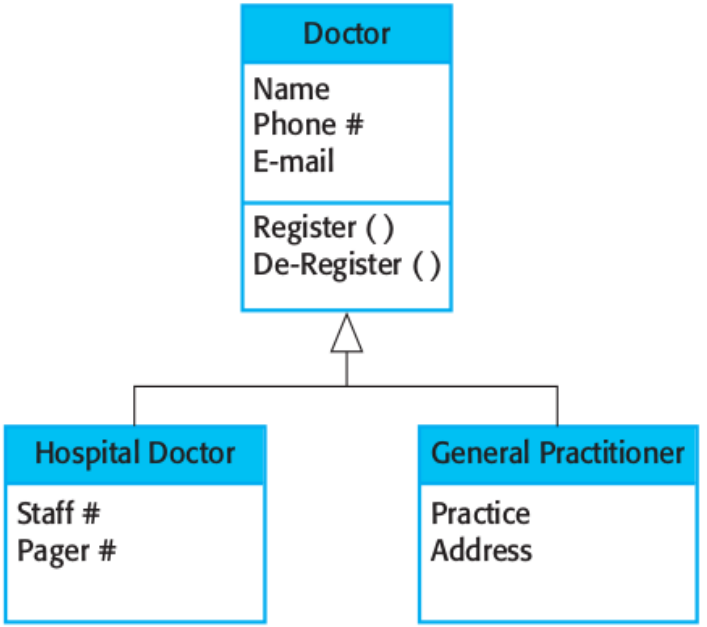
Sınıf Diyagramları



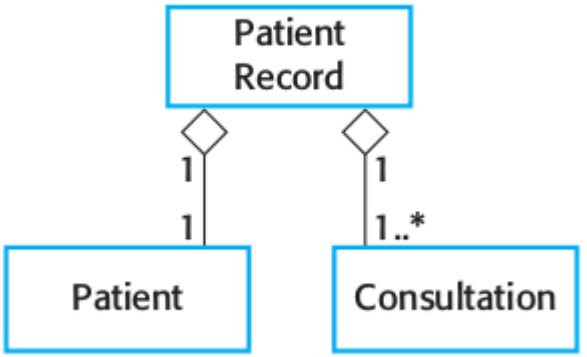


Genelleme


Doctor
Hospital Doctor
General Practitioner
Consultant
Team Doctor
Trainee Doctor
Qualified Doctor


Doctor
Name
Phone #
E-mail
Register ( )
De-Register ( )
Hospital Doctor
Staff #
Pager #
General Practitioner
Practice
Address

Bütünleştirme



Davranışsal Modeller

- Veri-güdümlü Modelleme
- Olay-güdümlü Modelleme

Veri Güdümlü Modelleme

- Veri güdümlü modeller veriyi işleme ve ilgili çıktıyı oluşturma sürecindeki olayların sırasını gösterirler.
- Gereksinim analizi esnasında, sistemin uçtan uca sürecini gösterdiği için yararlı olabilir
- İlk grafiksel yazılım modellri arasındadır. 1978'de DeMarco DFD(Data Flow Diagrams)'yi tanıtmıştır
- Veri akış modelleri belirli bir süreçle ilişkili verinin sistemde nasıl hareket ettiğini gösterdiklerinden analiz ve tasarım uzmanlarına yardımcıdır
- Veri akış diyagramları basit ve etkilidir, sistemin kullanıcılarına açıklanabilir düzeydedir
- UML veri akış diyagramlarını desteklemez. Bunun sebebi akış diyagramlarının sistem fonksiyonlarına odaklanıp sistem nesnelerini gözardı etmesidir.
- Veri güdümlü sistemler iş dünyasında çok yaygn olduğundan UML 2.0'da Sıralama Diyagramları eklenmiştir ve bunlar veri akış diyagramlarına benzerler

Olay Güdümlü Modelleme

- Sistemin iç veya dış olaylara nasıl cevap verdiğini gösterir
- Sistemde sonlu sayıda durum olduğu ve bir durumdan diğerine geçişin olaylarla (stimuli) gerçekleştiği varsayımına dayanır.
- Özellikle gerçek zamanlı sistemler için uygundur

Olay Güdümlü Modelleme

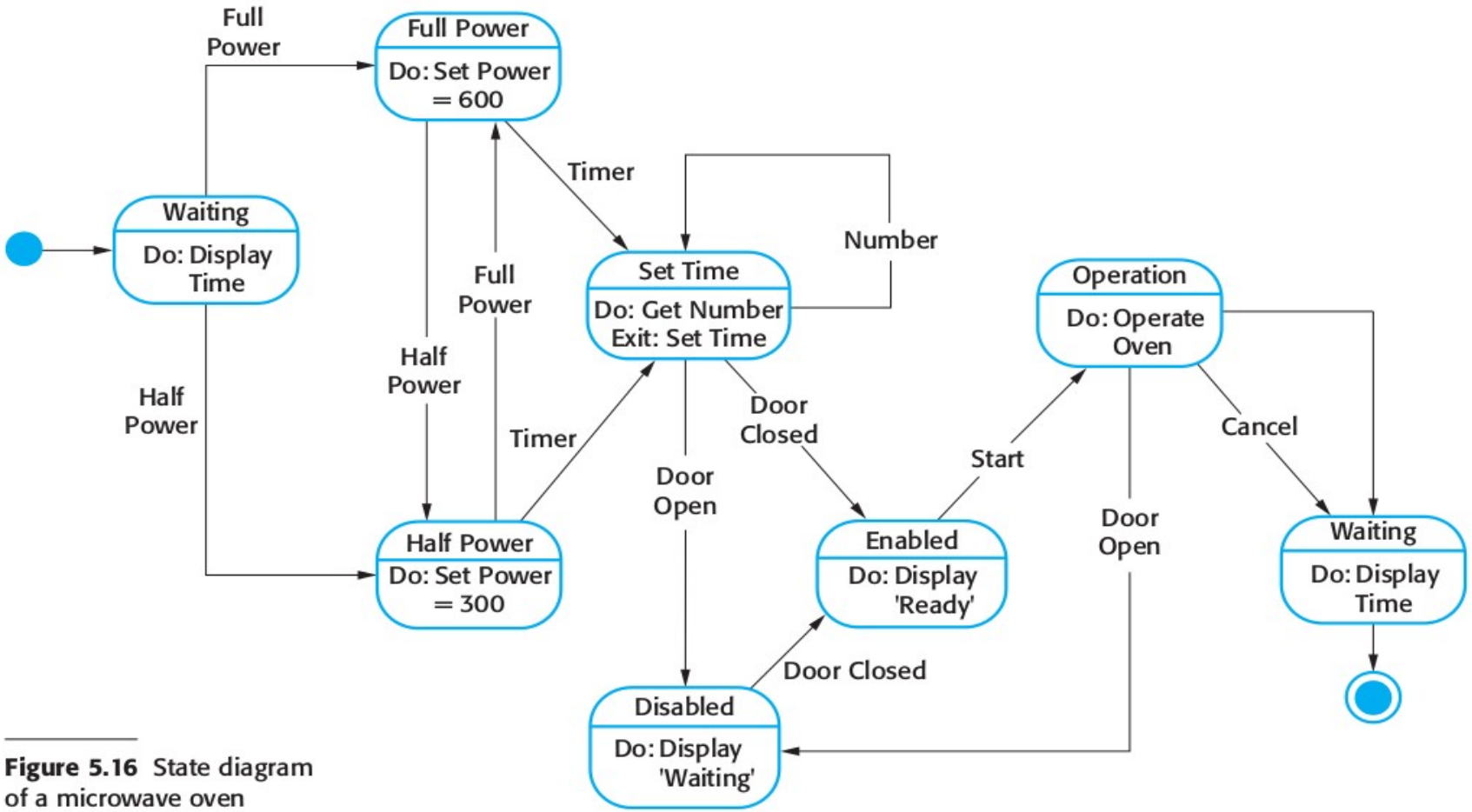


Figure 5.16 State diagram of a microwave oven

Model Tabanlı Mühendislik

- Temel çıktıların programlardan ziyade, modeller olduğu yazılım geliştirme yaklaşımıdır
- Çalışan yazılımlar bu modellerden otomatik olarak üretilirler
- Bu yaklaşımın destekçileri, artan soyutlama düzeyleri sayesinde mühendislerin programlama dillerinin detaylarıyla uğraşmak zorunda kalmayacaklarını söylerler

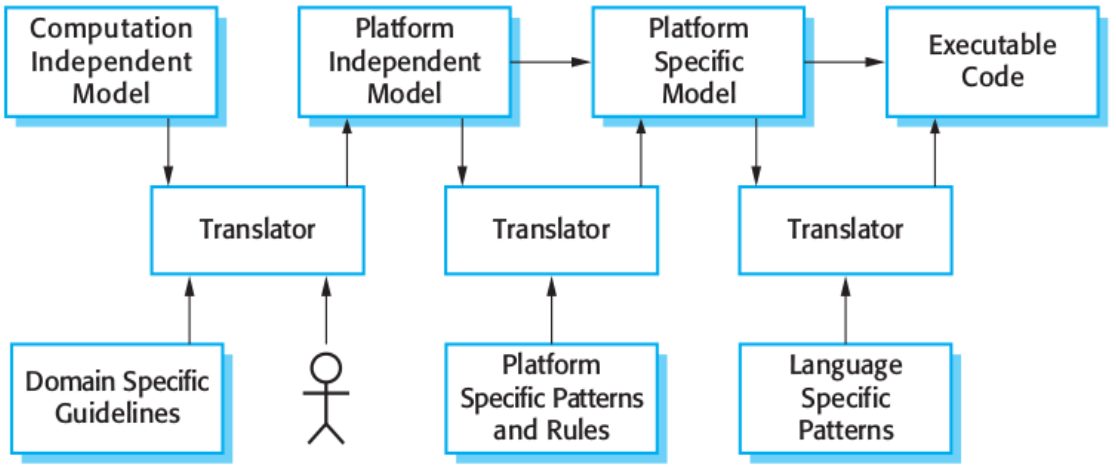
Model Tabanlı Mühendislik

- 2001'den beri kullanılmasına rağmen model tabanlı mimari, gelişiminin çok başındadır ve yazılım mühendisliğine büyük bir katkı yapıp yapmadığı belirsizdir
- Model Tabanlı Mühendislikle ilgili olumlu ve olumsuz temel görüşler:
  - Olumlu - Mühendisler sistemlerle ilgili, gerçekleştirimin detaylarından bağımsız olarak daha üst seviyede düşünürler. Bu olası hataları azaltır ve tasarım ve geliştirimi hızlandırır. Yeniden kullanılabilir, platformdan bağımsız uygulama modellerinin geliştirilmesini sağlar. Güçlü araçlar kullanarak aynı modelden farklı platformlar için sistem gerçekleştirimi yapılabilir. Sistemi yeni bir platforma uyarlamak için sadece o platform için bir çevirici yazmak yeterlidir.
  - Olumsuz - Model kullanımı, yazılım tasarımını tartışmak için iyi bir yöntemdir. Fakat modelde belirtilen soyutlamalar, gerçekleştirim kısmı için doğru olmayabilir. Dahası platform bağımsızlığı ile ilgili argümanlar ancak platformların yazılımın yaşam süresi boyunca eskiyeceği, çok uzun süre kullanılan büyük uygulamalar için geçerlidir. Fakat bu tip uygulamalar için de en hayati sorun gerçekleştirim değil, gereksinim mühendisliği, güvenlik, eski sistemle entegrasyon ve testtir.

Model Tabanlı Mimari

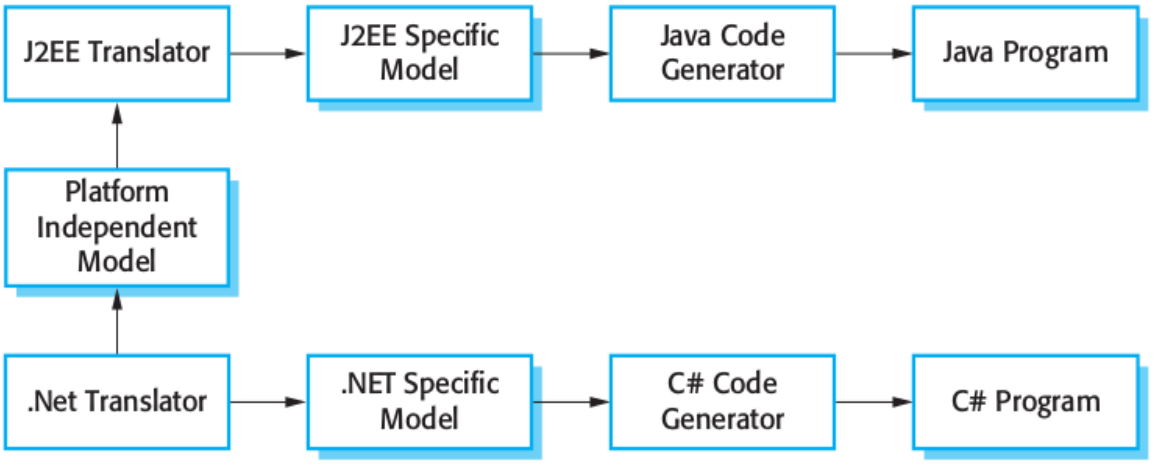
- Sistemi tanımlamak için UML modellerinin bir alt kümesini kullanan, yazılım tasarım ve geliştirimi için model odaklı bir yaklaşımdır
- Üç soyut sistem modelini önerir:
  - CIM
  - PIM (Platform Independent Model)
  - PSM (Platform Specific Model)

Model Tabanlı Mimari



←Platform Independent

Platform Specific →



Ödev

- **8 Aralık** ödev teslimi için son gün
- Ödevleriniz çıktı şeklinde teslim edilecek
- Yaptığınız bitirme projesi ile ilgili gereksinim ve temel tasarım dokümanı hazırlamanız beklenmektedir
- Bitirme projesi yapmayanlar herhangi bir proje için gereksinim hazırlayabilirler (stok takip, otomasyon, vb.)
- Gereksinim dokümanı hazırlama ile ilgili bilgilere, size verilen dokümanları okuyarak ve araştırma yaparak ulaşabilirsiniz
- Tasarımı UML diyagramlarıyla gösteriniz

Haftaya görüşmek üzere