

Chapter 5 – Sistem Modelleme

Lecture 1

Başlıklar



- ✧ İçerik/Bağlam (Context) modelleri
- ✧ Etkileşim Modelleri
- ✧ Yapısal Modeller
- ✧ Davranışsal Modeller
- ✧ Model Tabanlı Mühendislik

Sistem Modelleme



- ✧ Sistem modelleme, sistemin özet modellerinin geliştirilmesi sürecidir ve her bir model sistemi farklı bakış açıları ile gösterir.
- ✧ Genelde grafiksel gösterim ile yapılır.
 - Unified Modeling Language (UML).

Var olan ve Planlanan Sistemlerin Modelleri



- ✧ Var olan sistemlerin modelleri, gereksinim mühendisliği aşamasında kullanılır.
- ✧ Yeni sistemlerin modelleri, gereksinim mühendisliği aşamasında sistem gereksinimlerinin paydaşlara anlatılmasında kullanılabilir. Mühendisler bu modelleri gerçekleştirim için bir doküman olarak kullanır.
- ✧ Model tabanlı yaklaşımda, sistemin tamamının veya bir kısmının modeller üzerinde gerçekleştirilmesi mümkündür.

Sistemin bakış açıları



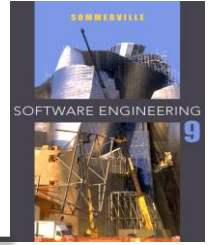
- ✧ İçerik/Bağlam bakış açısı: sistem nerde?
- ✧ Etkileşim bakış açısı: sistem çevresiyle nasıl etkileşiyor?
Sistemin bileşenleri nasıl etkileşiyor?
- ✧ Yapısal bakış açısı: sistemin veya sistemin işlediği verilerin organizasyonunu nasıl?
- ✧ Davranışsal bakı açısı: sistemin dinamik davranışları neler ve olaylara nasıl yanıt verir?

UML diyagram tipleri



- ✧ Aktivite diyagramları
- ✧ Kullanım durumu diyagramları
- ✧ Sequence diyagramları
- ✧ Sınıf diyagramları
- ✧ Durum diyagramları

İçerik Modelleri



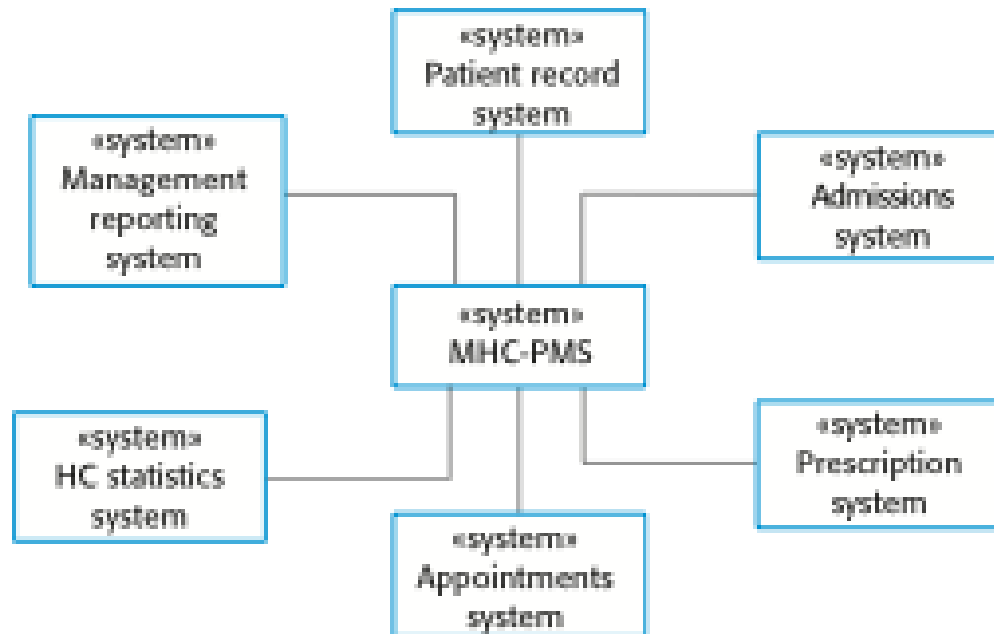
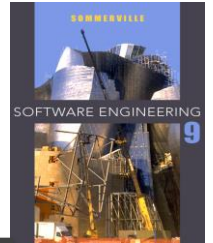
- ✧ Operasyonel çevreyi gösterir
- ✧ Sosyal ve organizasyonel gereksinimleri sistemin sınırlarını belirleyebilir.
- ✧ Mimari modelleri, sistemin diğer sistemlere göre olan konumunu gösterir.

Sistem Sınırları



- ✧ Sistemin içinde ne var? Sistemin dışında ne var?
 - Sistemin geliştirilmesi esnasında kullanılan diğer sistemler
- ✧ Sistemin sınırları bir pazarlık konusu olabilir.

MHC-PMS'nin Bağlamı

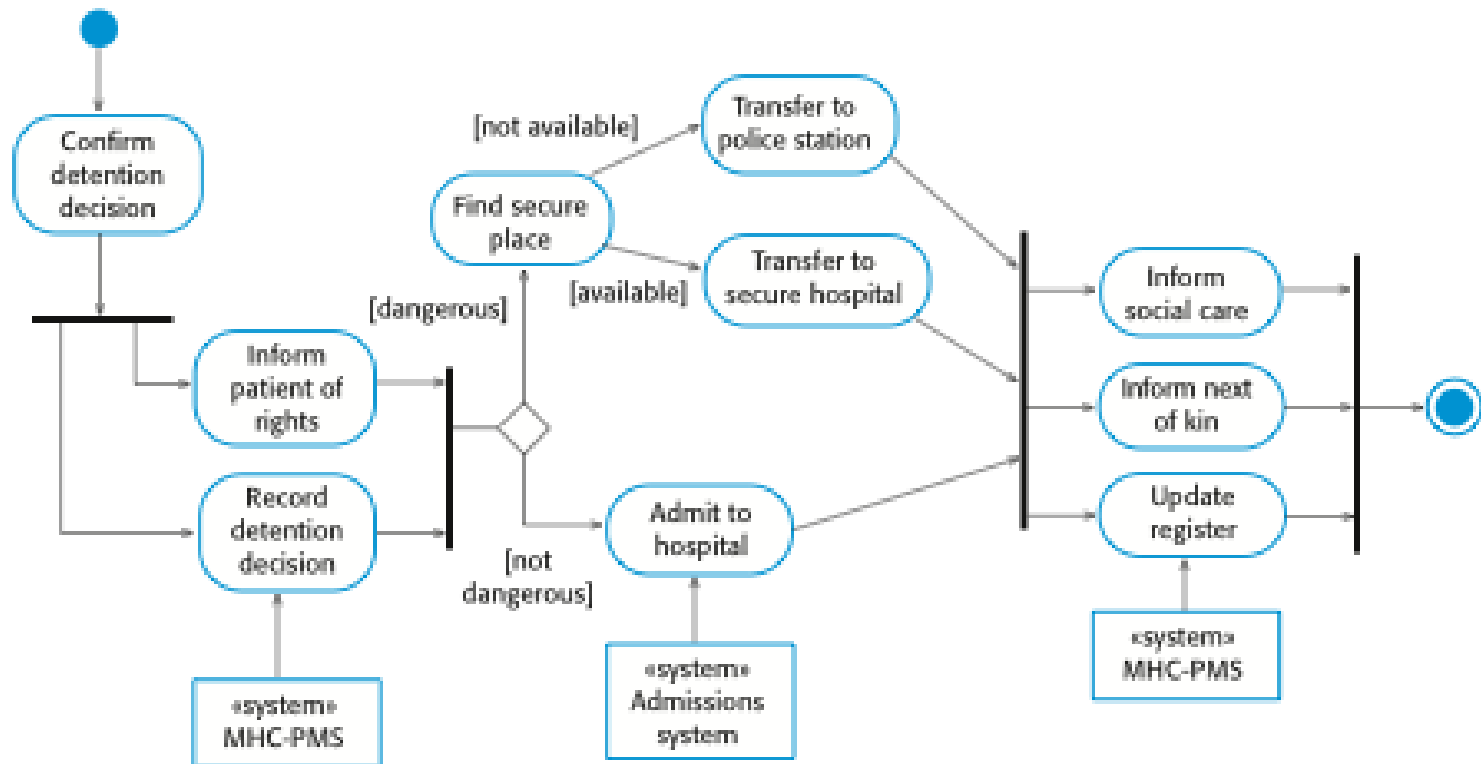
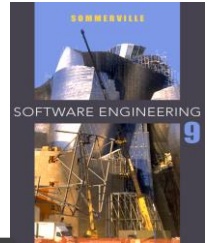


Süreç Bakış Açısı



- ✧ Bağlam modelleri çevreyi gösterir, etkileşimi değil!
- ✧ UML aktivite diyagramları kullanılabilir.

Alıkoyma Süreç Modeli

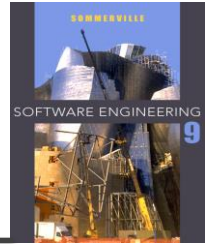


Etkileşim Modelleri



- ✧ Kullanıcı etkileşimlerini göstermek, kullanıcı gereksinimlerini belirlemek için önemli.
- ✧ Sistemler arası etkileşimleri modellemek, iletişim problemlerinin görülmesini sağlar.
- ✧ Etkileşimler, performans ve bağımlılık ile ilgili bilgi verir.
- ✧ Kullanım durumu ve sequence diyagramları kullanılabilir.

Kullanım Durumu Modelleme



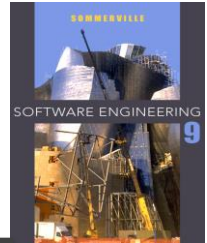
- ✧ Farklı görevler için farklı kullanım durumları oluşturulur.
- ✧ Aktörler, insan veya sistem olabilir.
- ✧ Desteklemek için «metinsel» içerik kullanılmalıdır.

Transfer-data Kullanım Durumu

✧ MHC-PMS'deki bir kullanım durumu



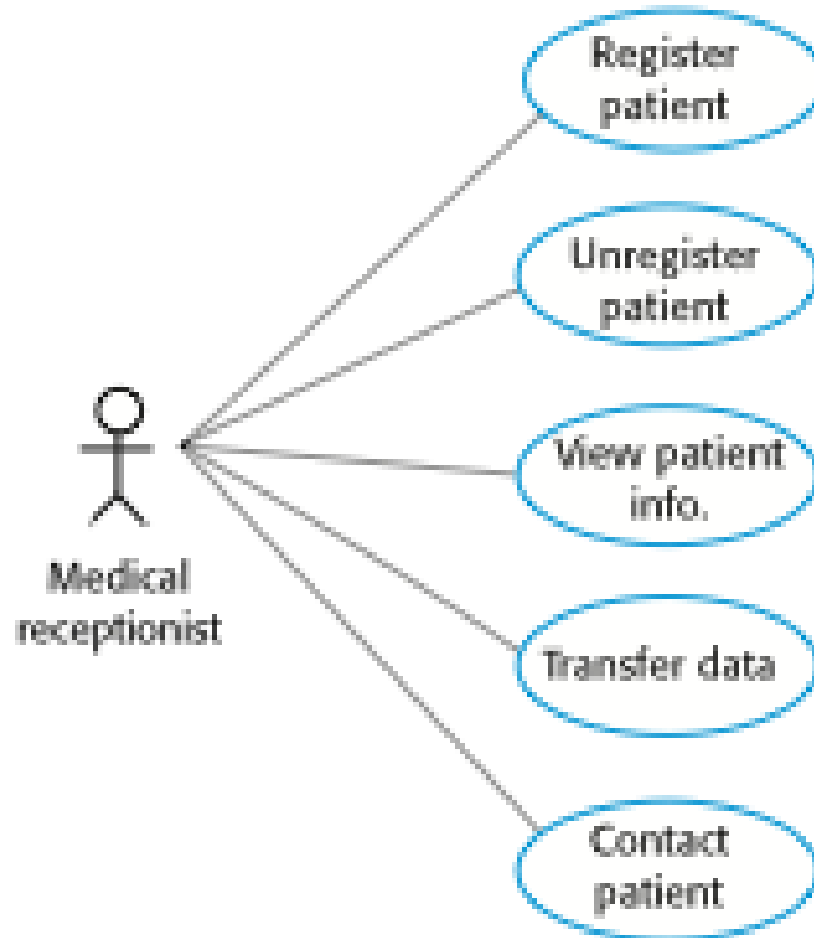
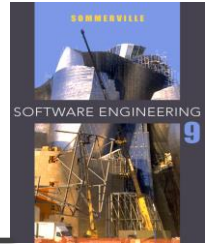
‘Transfer data’ kullanım durumunun «sekmeli» tanımı



MHC-PMS: Transfer data

Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary
Stimulus	User command issued by medical receptionist
Response	Confirmation that PRS has been updated
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

MHC-PMS'deki «hasta kayıt görevlisinin» bulunduğu kullanım durumları

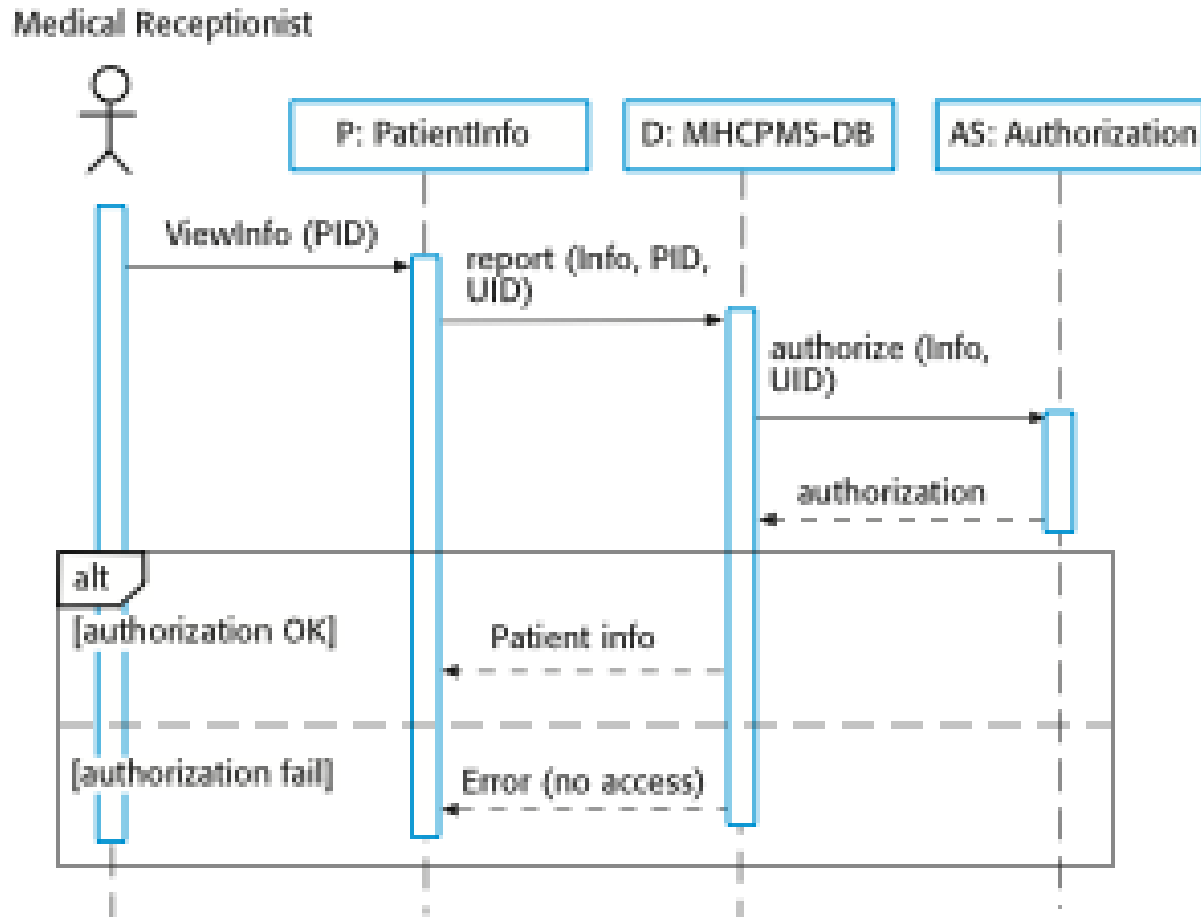
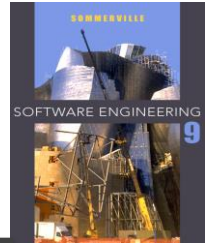


Sequence diyagramları



- ✧ Aktörler ve sistemdeki nesneler arasındaki etkileşimi göstermek için kullanılır
- ✧ Bir sequence diyagramı, bir kullanım durumundaki (veya bir kullanım durumu parçasındaki) etkileşimlerin sırasını gösterir.

Sequence diagram – Hasta bilgisi görme



Yapısal Modeller



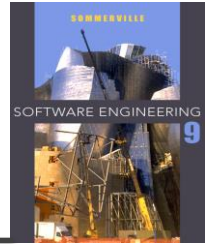
- ✧ Yapısal modeller, sistemi oluşturan bileşenleri ve bunların ilişkilerini gösterir.
- ✧ Yapısal model statik de olabilir dinamik de olabilir.
- ✧ Sistem mimarisini tartışırken yapısal model oluşturmak faydalıdır.

Sınıf diyagramları

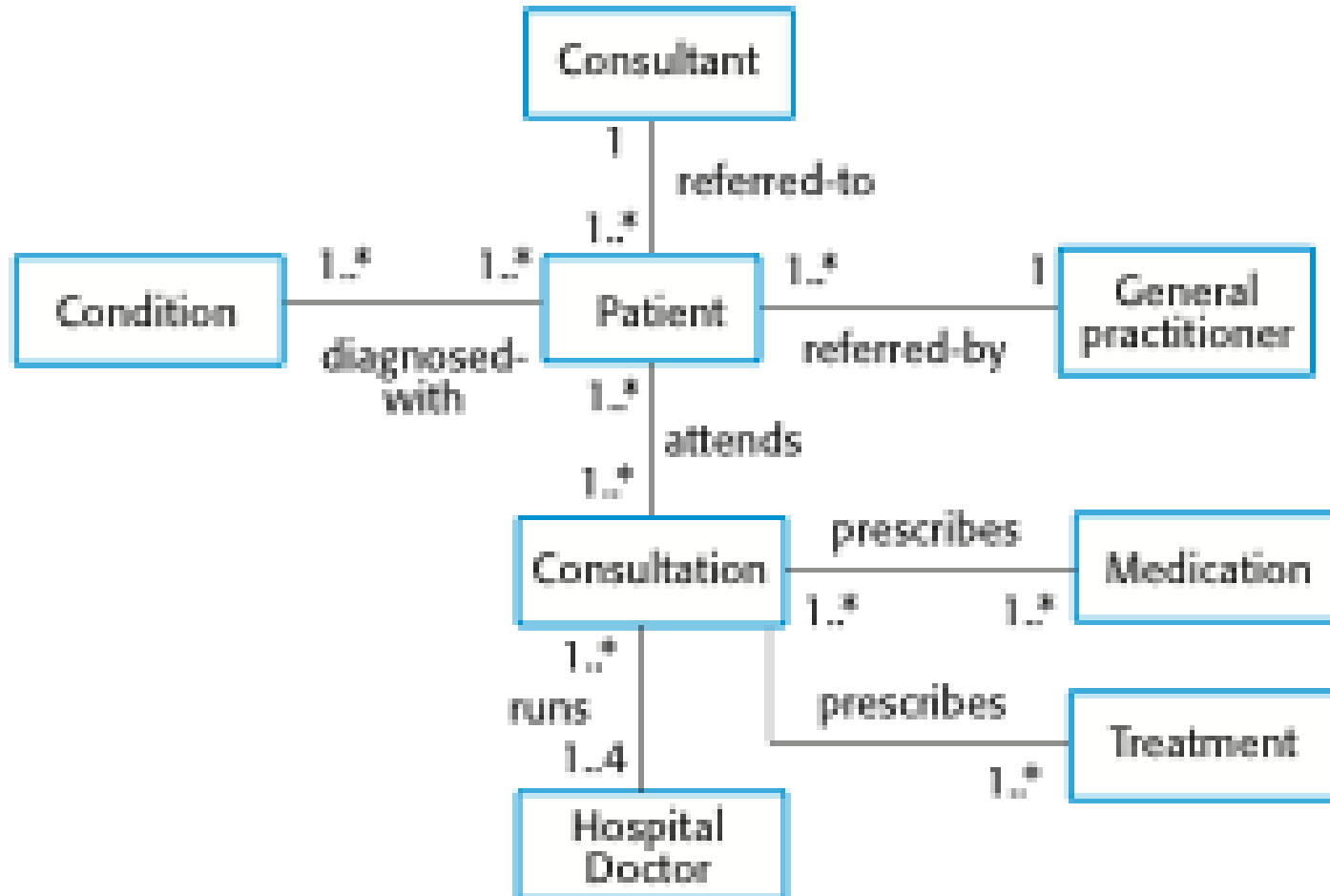
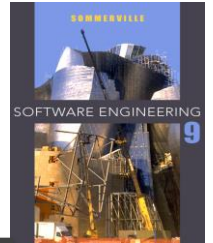


- ✧ Sınıf, bir sistem nesnesinin «genel» tanımıdır.
- ✧ Nesneler sıklıkla gerçek dünyadaki «şeyleri» gösterir: hasta, ilaç, reçete, doktor ...

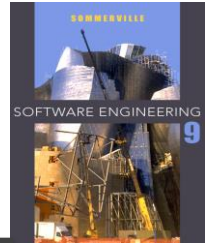
UML sınıfları ve ilişkileri



MHC-PMS'deki sınıflar ve ilişkileri



Konsültasyon Sınıfı



Consultation

Doctors
Date
Time
Clinic
Reason
Medication prescribed
Treatment prescribed
Voice notes
Transcript

...

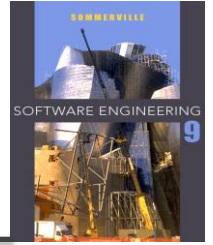
New ()
Prescribe ()
RecordNotes ()
Transcribe ()

...

Chapter 5 – Sistem Modelleme

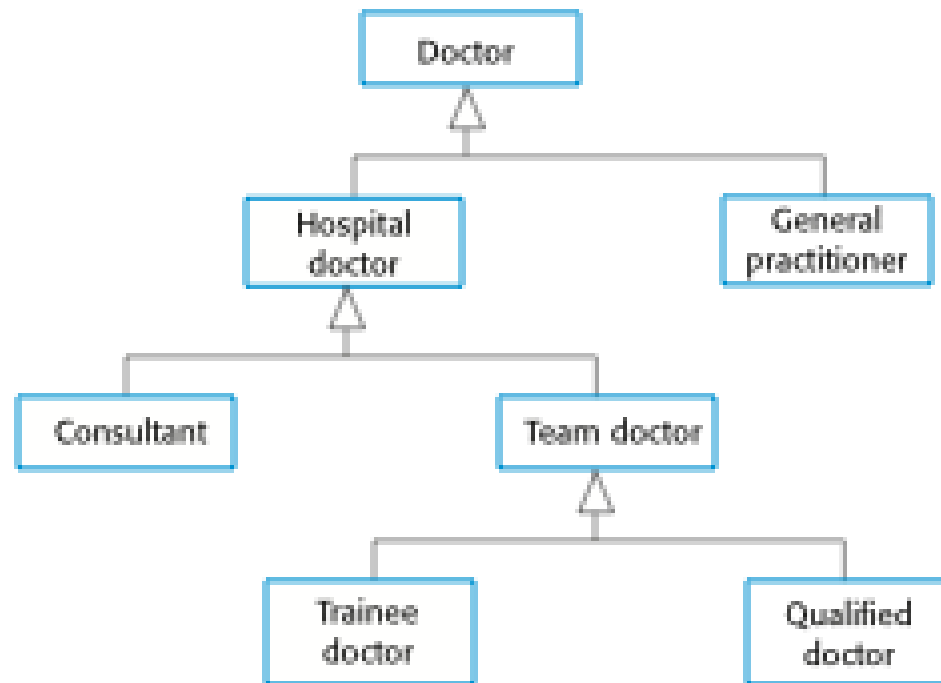
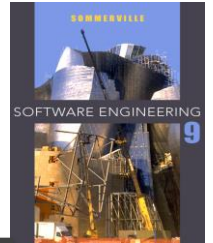
Lecture 2

Genelleme

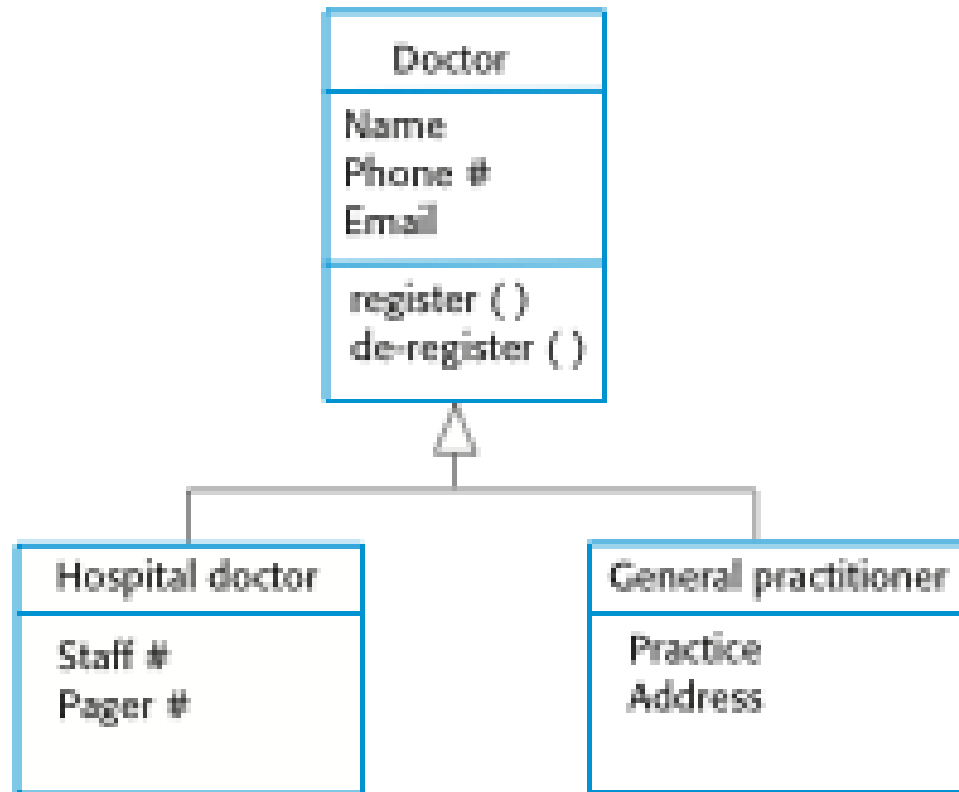
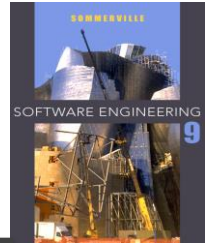


- ✧ Karmaşıklığı yönetmek için kullanılan bir yöntem
- ✧ Varlıkları sınıflandırırız. (hayvanlar, otomobilleri vb.)
- ✧ Aynı sınıftaki üyelerin benzer karakteristiklere sahip olduklarını belirtir.
- ✧ Nesneye yönelimli dillerde genelleme, sınıf kalıtım mekanizması ile oluşturulur.

Bir Genelleme Hiyerarşisi



Daha Detaylı Bir Genelleme Hiyerarşisi



Davranışsal Modeller



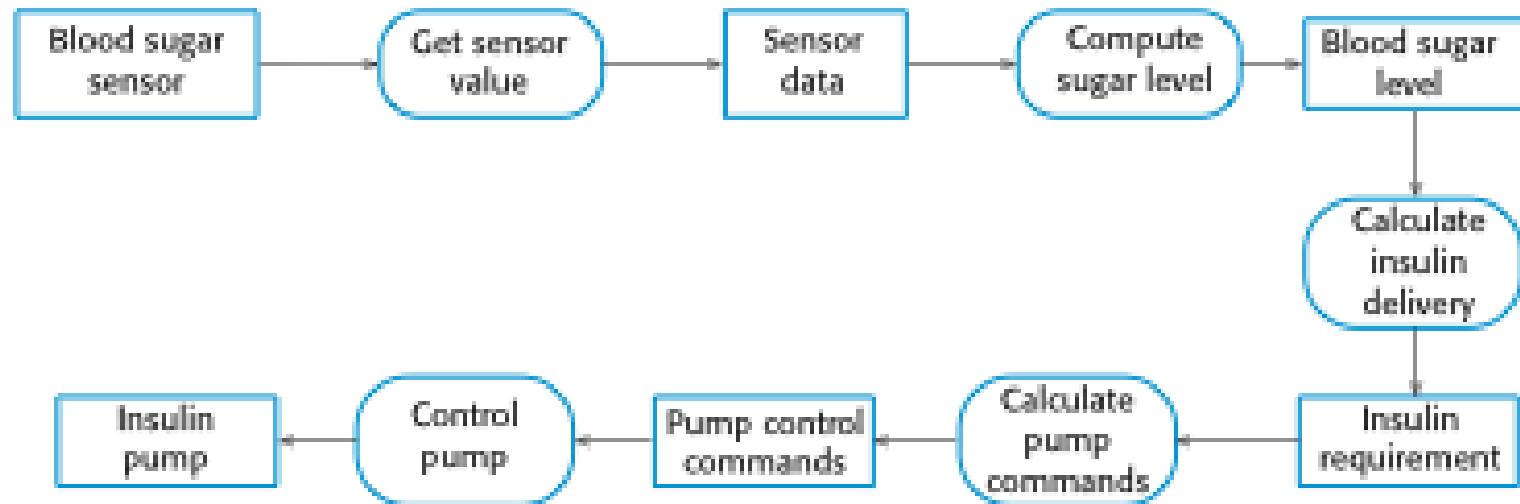
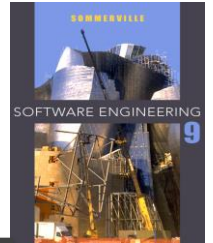
- ✧ Davranışsal modeller, bir sistemi çalışıyormuş gibi dinamik olarak modeller.
- ✧ Çevreden gelen bir «uyarana» sistemin nasıl cevap verdiğini gösterir.
- ✧ Uyarılar 2 tiptir
 - Veri
 - Olaylar

Veriye Dayalı Modelleme

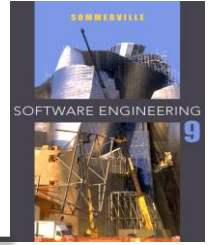


✧ Sistemin ne yapacağını sisteme gelen veriler belirler.

İnsülin pompası'nın çalışması için bir aktivite modeli



Olaya Dayalı Modelleme



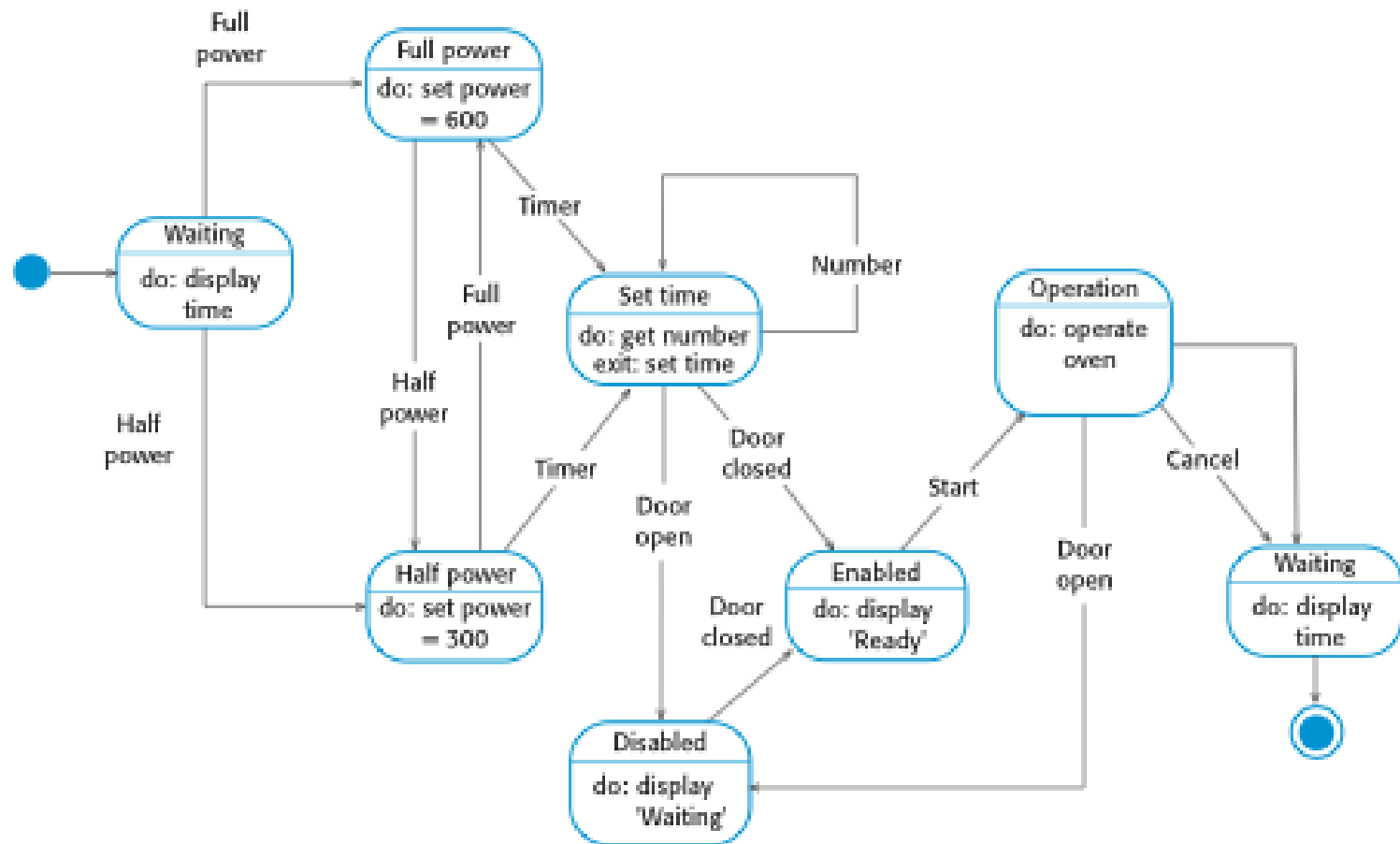
- ✧ Gerçek zamanlı sistemler çoğunlukla olaya dayalı sistemlerdir ve veri işleme gereksinimi en azdır.
- ✧ Olaya dayalı bir model, sistemin içsel ve dışsal olaylara nasıl yanıt verdiğini gösterir.
- ✧ Durumlar ve durumları değiştiren olaylar olarak düşünülebilir.

Durum Makinesi Modelleri

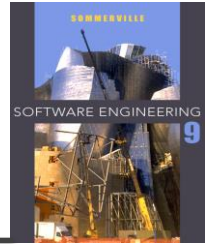


- ✧ Durum makinesi modellerinde durumlar düğümler olarak ve olaylar da bu düğümleri bağlayan kenarlar olarak gösterilir.
- ✧ Bir olay olduğunda, sistem bir durumdan başka bir duruma geçer.

Mikroalga Fırın İçin Bir Durum Modeli



Mikrodalga Fırın İçin «Uyaranlar»



Uyaran	Tanım
Half power	Kullanıcı half-power düğmesine bastı
Full power	Kullanıcı full-power düğmesine bastı
Timer	Kullanıcı timer düğmelerinden birine bastı
Number	Kullanıcı sayı düğmelerine bastı
Door open	Fırının kapı kilidi kapalı değil
Door closed	Fırının kapı kilidi kapalı
Start	Kullanıcı Start düğmesine bastı
Cancel	Kullanıcı Cancel düğmesine bastı

Model Tabanlı Mühendislik



- ✧ Model tabanlı mühendislik, geliştirme sürecinin çıktılarını (çalıştırılabilir yapıyı) «programlamaktan» ziyade «modellemeye» dayanır.
- ✧ Donanım/Yazılım platformları üzerinde çalışacak olan programlar doğrudan modellerden üretilir.

Model Tabanlı Mühendislik



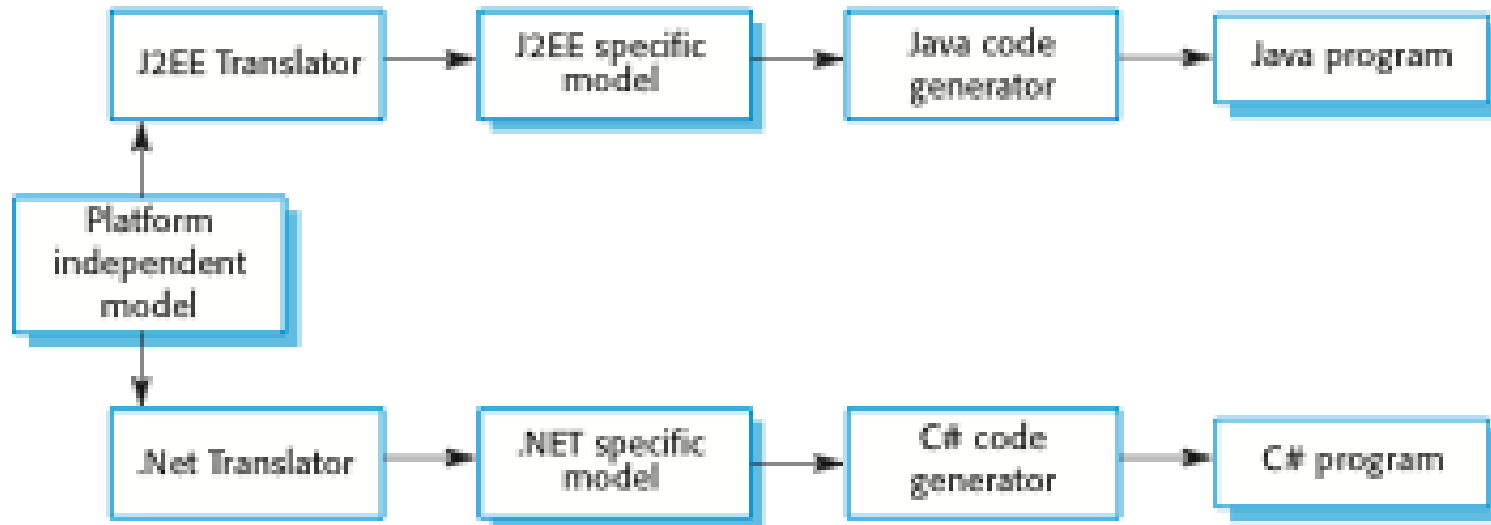
- ✧ Model tabanlı mühendislik yaklaşımı halen tam olarak olgunlaşmadı.
- ✧ Avantajları
 - Sistemin yüksek seviyelerde soyut olarak gösterilmesini sağlar
 - Otomatik kod üretimi süreci ucuzlatır
- ✧ Dezavantajları
 - Modeller, gerçekleştirim için yeterince detaylı olmayabilir.
 - Yeni platformlar için «tercümanlar» geliştirme işi, kod yazmaktan daha zordur.

Model Tipleri

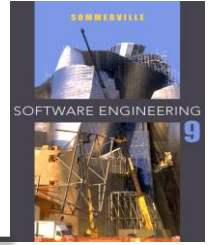


- ✧ A computation independent model (CIM)
 - These model the important domain abstractions used in a system. CIMs are sometimes called domain models.
- ✧ A platform independent model (PIM)
 - These model the operation of the system without reference to its implementation. The PIM is usually described using UML models that show the static system structure and how it responds to external and internal events.
- ✧ Platform specific models (PSM)
 - These are transformations of the platform-independent model with a separate PSM for each application platform. In principle, there may be layers of PSM, with each layer adding some platform-specific detail.

Çoklu platforma özgü modeller



Çevik yöntemler ve Model Tabanlı Mühendislik



- ✧ Model tabanlı yaklaşımın geliştiricileri, artırılmış geliştirmeyi desteklediği için bu yaklaşımın çevik yöntemlerle kullanılabileceğini iddia ederler.
- ✧ Ancak, geliştirme öncesi yapılan ön modellemelerin çokluğu çevik manifesto ile çelişir.
- ✧ Eğer bütün süreç tam olarak otomatikleştirilebilirse (fazladan kod yazmaya gerek kalmazsa) model tabanlı yaklaşım çevik yöntemlerle kullanılabilir.

Çalıştırılabilir UML



- ✧ Executable UML veya xUML
- ✧ Grafiksel gösterimle birlikte «zamanlama» ve «semantik» kuralları içerir.
- ✧ Araçlar
 - IBM Rational Software Architect Simulation Toolkit