

Mühendislik Projesi Engineering Project

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Ralph M. Ford, Chris S. Coulston, Design for Electrical and Computer Engineers, McGraw Hill, 2008." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ Veri akış diyagramları
- ▶ Varlık ilişki diyagramları
- ▶ UML

Modeller

- ▶ Fonksiyonel ayrıştırma, giriş, çıkış ve davranış geçişini modellemek için iyi bir araçtır.
- ▶ Ancak, bir **sistemin durum diyagramları, akış diyagramları, veri akış diyagramları ve varlık ilişki diyagramlarının oluşturulması gerekli olabilir.**
- ▶ **Bir model**, bir **sistemin, sürecin veya nesnenin standartlaştırılmış sunumudur** ve fiziksel gerçekleştirilmesine yönelik bilgi içermez.
- ▶ **Modelleme dilleri** kelimeler veya cümleler yerine **grafik sembolleri kullanır.**

3

Modeller

- ▶ Bir model **aşğıdaki özellikleri sağlamalıdır:**
 - ▶ **Soyut olmalıdır.** Model fiziksel gerçekleştirilmeden bağımsız olmalıdır.
 - ▶ **Belirsiz olmamalıdır.** Model isteneni açık ve net ifade etmelidir.
 - ▶ **Yenilikçiliğe izin vermelidir.** Alternatif sistem gerçekleştirmek mümkün olmalıdır.
 - ▶ **Standart olmalıdır.** Herkes tarafından anlaşılır standart bir dile sahip olmalıdır.
 - ▶ **Değiştirilebilir olmalıdır.** Bir modelde modifikasyon yapılabilmelidir.
 - ▶ **Detay içermemelidir.** Gereksiz detaylar içermemelidir ve önemli özellikleri vurgulamalıdır.
 - ▶ **Problemi parçalara bölmelidir.**

4

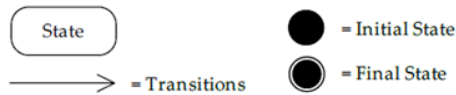
İçerik

- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ Veri akış diyagramları
- ▶ Varlık ilişki diyagramları
- ▶ UML

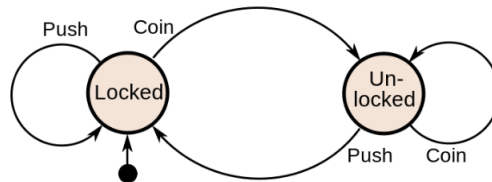
5

Durum diyagramları

- ▶ Durum diyagramları (**state diagrams**), bir **sistemin davranışını hafıza ile tanımlar** (geçmiş bilgisini içerir).
- ▶ **Hafızası olan bir sistem**, aldığı girişe vereceği **cevabı bulunduğu duruma göre belirler**.
- ▶ Eğer bir sistem **aynı giriş için farklı çıkış üretiyorsa** hafızası vardır ve **durum diyagramları ile modellenir**.



- ▶ **Örnek:** Turnike kontrol sistemi



6




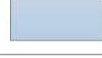

İçerik

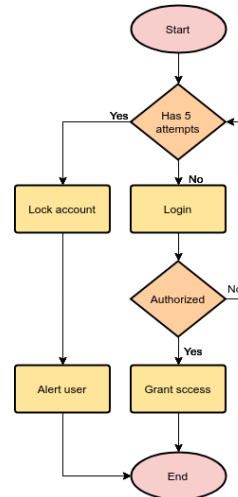
- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ Veri akış diyagramları
- ▶ Varlık ilişki diyagramları
- ▶ UML

7

Akış diyagramları

- ▶ Akış diyagramları (**flowcharts**), bir algoritmanın veya sürecin adımlarını görsel olarak tanımlar.
- ▶ **Yaygın kullanılır** ve temel **bilinen sembollere sahiptir**.

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision



8

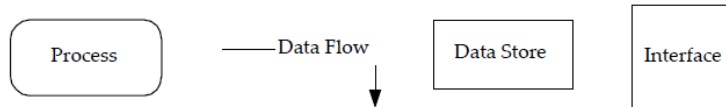
İçerik

- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ **Veri akış diyagramları**
- ▶ Varlık ilişki diyagramları
- ▶ UML

9

Veri akış diyagramları

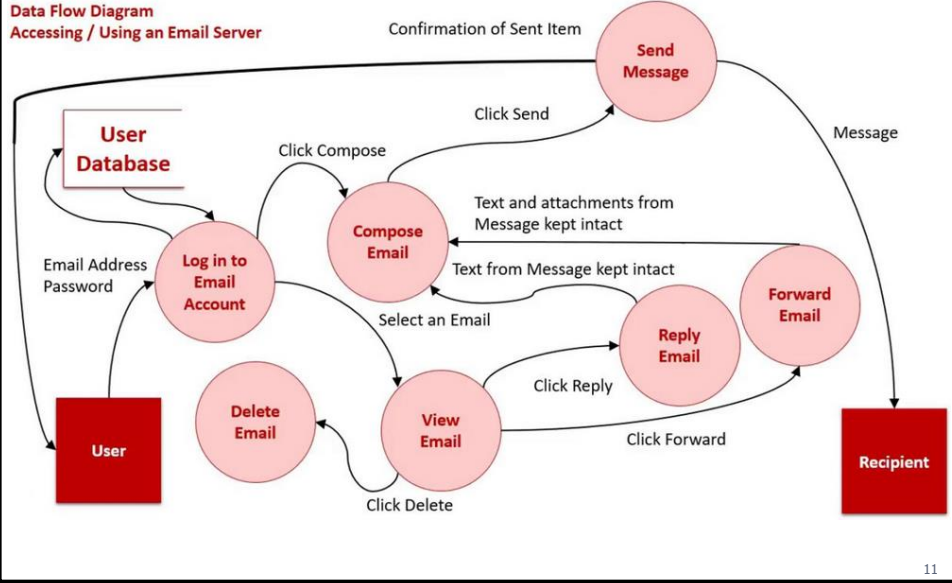
- ▶ Veri akış diyagramının **amacı verinin akışı ve işlenmesini modellemektir.**
- ▶ Veri akış diyagramı, **girişleri alır, dönüştürür ve çıkış verisini üretir.**
- ▶ **Data process:** Bir işlevi ya da görevi tanımlar.
- ▶ **Data flow:** İki process arasında veri ilişkisini gösterir.
- ▶ **Data store:** Veri saklamayı gösterir.
- ▶ **Interface:** Sistemi dışarıdan kullanan varlıkları tanımlar.



10

Veri akış diyagramları

Örnek: E-posta sunucuya erişim ve kullanım



İçerik

- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ Veri akış diyagramları
- ▶ Varlık ilişki diyagramları
- ▶ UML

Varlık ilişki diyagramları

- ▶ Veritabanı veriyi saklayan ve geri getiren bir sistemdir.
- ▶ Veritabanı varlık ilişki diyagramı ile modellenebilir.
- ▶ Varlık ilişki diyagramı ile **bir grup nesne, özellikleri ve aralarındaki ilişkiler gösterilir.**
- ▶ Varlık ilişki modelinde üç tür bileşen vardır:
 - ▶ **Varlıklar (entities):** Nesnelere, roller, organizasyonel birimler, cihazlar veya lokasyonlar olabilir. Bir varlık için örnek (instance) ifade edilebilir (Student, Kemal).
 - ▶ **İlişkiler (relationships):** Varlıklar arasındaki ilişkileri tanımlayıcıdır.
 - ▶ **Özellikler (attributes):** Varlıkların örnekleri arasındaki farkı oluşturan değerlerdir.

13

Varlık ilişki diyagramları

Örnek: Bir üniversite veritabanı için varlık ilişki diyagramı.

- ▶ Üniversite; **Öğrenciler, Dersler ve Bölümler** olarak üç varlığa sahiptir.
- ▶ Varlıklar arasında varlık ilişki matrisi oluşturulabilir.

	Student	Course	Department
Student		takes many	majors in one
Course	has many	can require many / can be the prerequisite for many	is offered by one
Department	enrolls many	offers many	

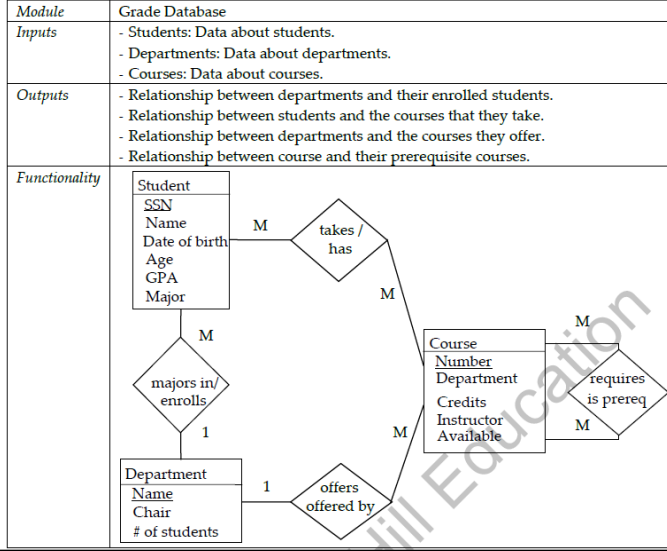
- ▶ **Student-Course** arasında **many-to-many (M:M)**, **Student-Department** arasında **many-to-one (M:1)** ilişkisi vardır.
- ▶ Bir ders çok sayıda ön şart gerektirebilir veya kendisi çok sayıda ders için ön şart olabilir.

14

Varlık ilişki diyagramları

Örnek: devam.

- ▶ Varlıklar arasında ilişki ve anahtar özellikler (key attributes).



15

İçerik

- ▶ Modeller
- ▶ Durum diyagramları
- ▶ Akış diyagramları
- ▶ Veri akış diyagramları
- ▶ Varlık ilişki diyagramları
- ▶ UML

16

UML

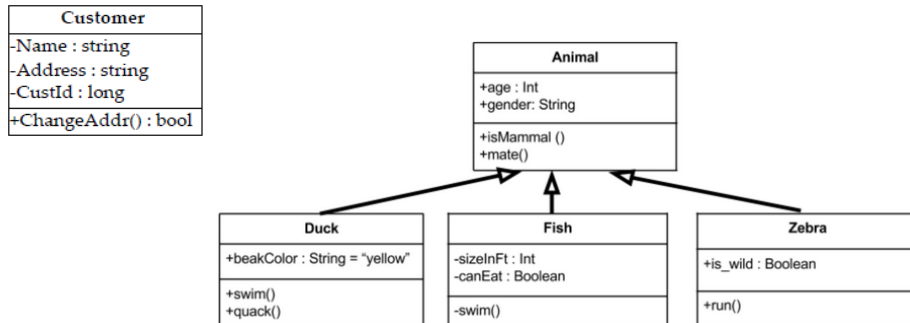
- ▶ **Unified Modeling Language (UML), nesne yönelimli yazılım geliştirme sürecini modellemek için geliştirilmiştir.**
- ▶ UML çok farklı sistem türlerinde de modelleme amacıyla kullanılmaktadır.
- ▶ UML bir sistem için farklı bakış açıları oluşturur:
 - ▶ **Static view**
 - ▶ **Use-case view**
 - ▶ **State machine view**
 - ▶ **Activity view**
 - ▶ **Interaction view**
 - ▶ **Physical view**

17

UML

Static view

- ▶ Nesne yönelimli programlamada bir **nesne** hem **veriyi** hem de **fonksiyonları içerir.**
- ▶ **Statik görünüm, bir sistemdeki sınıfları ve aralarındaki ilişkileri gösterir.**
- ▶ Statik görünüm sınıf diyagramını ifade eder.

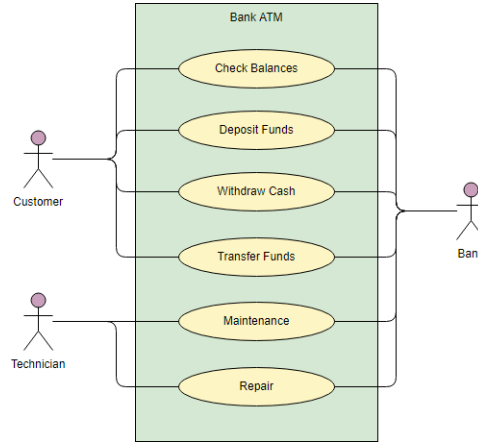


18

UML

Use-case view

- ▶ **Kullanıcı bakış açısıyla modellemek için kullanılır.**
- ▶ Kullanıcı bakış açısı **use-case** diyagramını ifade eder.
- ▶ Use-case diyagramı **aktör** ve **use-case sembollerine sahiptir.**

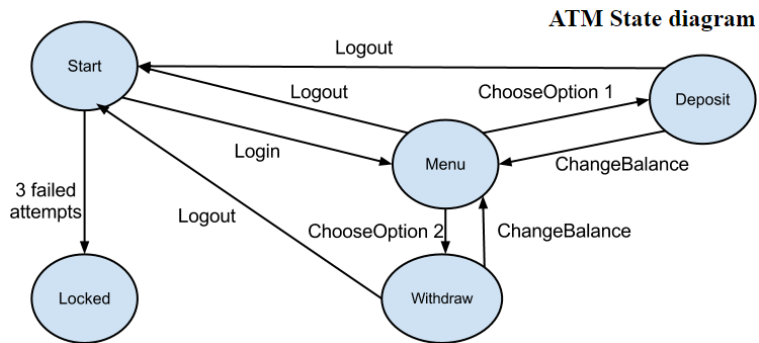


19

UML

State machine view

- ▶ **Durum makinesi ile sistemi modellemek için kullanılır.**
- ▶ Durum makinesi gösterimi durum diyagramını ifade eder.
- ▶ **Geçiş ve durum sembollerine sahiptir.**

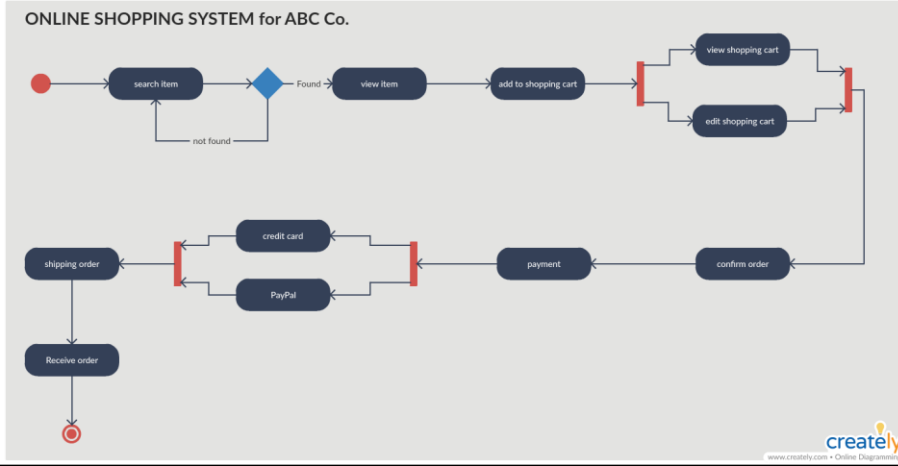


20

UML

Activity view

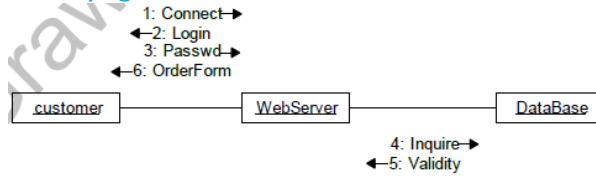
- ▶ Etkinlik görünümü bir işi tamamlamak için gereken adımları tanımlamak amacıyla kullanılır.
- ▶ Birleştirme, ayrıştırma, karar ve ekleme sembolleri kullanılabilir.



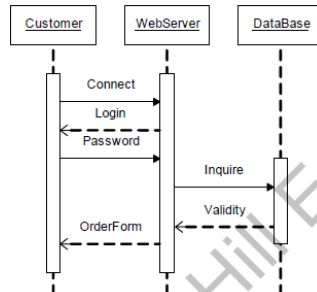
UML

Interaction view

- ▶ Nesnelar arasındaki etkileşimi gösterir.
- ▶ Collaboration diyagramları nesnelar arasındaki mesajları gösterir.



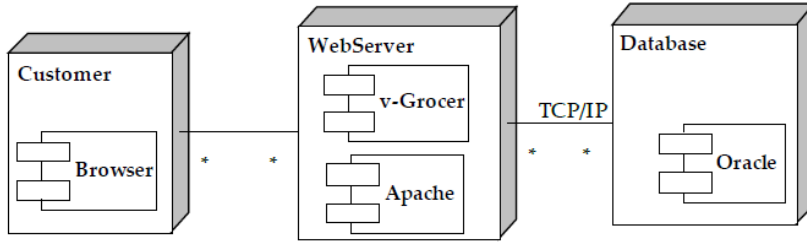
- ▶ Sequence diyagramları mesaj sırasını vurgular.



UML

Physical view

- ▶ **Sistemin fiziksel bileşenlerini göstermek için kullanılır.**
- ▶ **Component diyagramı**, yazılım dosyalarını gösterir.
- ▶ **Deployment diyagramı**, donanım ve bileşenlerin haberleşmesini gösterir.
- ▶ Şekilde, **Customer**, **WebServer** ve **Database** donanımsal bileşenleri; **Browser**, **v-Grocer**, **Apache** ve **Oracle** yazılım bileşenlerini gösterir.



23

Ödev

- ▶ Görme engelli bireyler için dış ortamlarda kullanılmak üzere geliştirilecek bir mobil asistan cihazı için aşağıdakileri oluşturunuz:
 - ▶ Durum diyagramı
 - ▶ Akış diyagramı
 - ▶ Veri akış diyagramı
 - ▶ Varlık ilişki diyagramı
 - ▶ UML görünümleri

24