

# Sistem Analizi ve Tasarımı

5.Ders

Göksel Biricik

## Bu Derste

- Sistem Analizi
- Fonksiyonel Çözümleme
- SRS
- Kavramsal Modelin Oluşturulması

## Gereksinim Analizi

- Sistem analizi modelinin amacı:
  - İhtiyaçları açıklamak
  - Tasarımın nasıl oluşturulacağını temelini oluşturmak
  - Oluşturulan yazılımın ihtiyaçları karşılayıp karşılamadığını onaylayan unsurları belirlemek

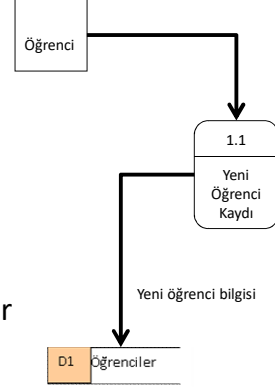
## Gereksinim Analizi Modeli

- Veri Sözlüğü: Tüm veri nesnelerinin tanımları (metadata)
- Varlık İlişki Diyagramı (ERD): Veri nesneleri arası ilişkiler
- Veri Akış Diyagramı (VAD): Verilerin nasıl taşındığı, veri akışını sağlayan fonksiyonların neler olduğu
- Durum Geçiş Diyagramı (STD): Sistem dışındaki olaylar sonucunda sistemin nasıl hareket ettiği



## Veri Akış Diyagramı

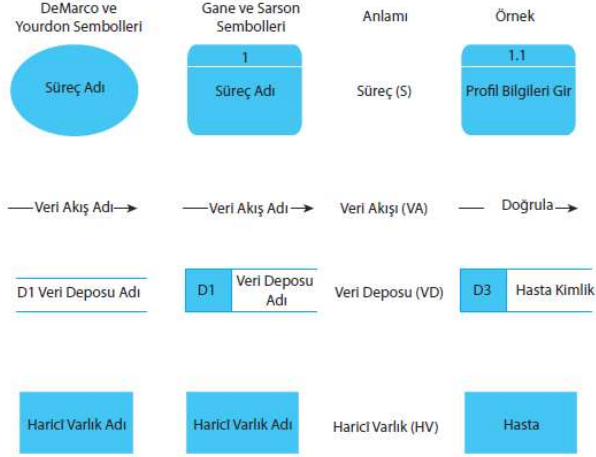
- Sistemdeki
  - Varlıklar
  - Süreçler - işlemler
  - Veri depoları
  - Aralarında verinin akışı
- Birden fazla varlık olabilir
- Okların çakışmaması için aynı varlık tekrarlayabilir
- Oklar tek yönlüdür
- Okların tek bir kaynağı ve hedefi vardır
- İşlemler hiyerarşiye uygun numaralandırılır



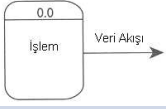
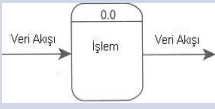
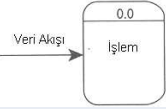
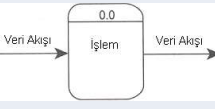

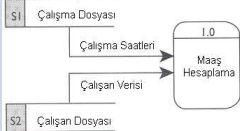
## VAD Öğeleri

- **Varlıklar:** Kişi, kurum, birim, sistem olabilir. Birincil aktörlere karşılık gelir. Sisteme veri sağlar ya da veri alır.
- **Veri Akışı:** sistemde bir yerden başka bir yere hareket eden veri (ör. barkod no) ya da mantıksal veri koleksiyonu (ör. Rapor çıktısı içeriği).
- **Süreç:** Belirli bir işi gerçekleştirmek amacıyla elle veya bilgisayarla yürütülen etkinlik/fonksiyon. Emir kipinde yükleme isimlendirilir. (ör. Randevu al) Her süreçte tek etkinlik gerçekleşir.
- **Veri Deposu:** Verilerin kalıcı olarak bulunduğu yerler (dosya, klasör, veritabanı, form, çıktı, rapor, karekod, ...)

## VAD Sembolleri



## VAD Kuralları

Kural	Yanlış	Doğru
İşlemin sadece çıkışı olamaz.		
İşlemin sadece girişi olamaz.		
İşlem girişleri istenen çıkışı verecek kadar yeterli olmalıdır.		

## VAD Kuralları

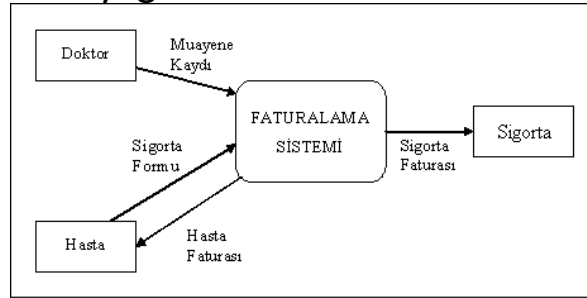
Kural	Yanlış	Doğru
Her veri deposu bir işlemle ilgili olmalıdır.		
Veri deposu bir varlıkla doğrudan ilişkide olamaz.		
Veri akışı oku iki yönlü olamaz. Bir işlemle veri deposu arasında karşılıklı veri akışı varsa farklı tek yönlü oklarla gösterilmelidir.		

## VAD Kuralları

Kural	Yanlış	Doğru
Bir işlemden farklı iki işleme gidecek olan aynı veri, aynı yönde iki uçlu okla gösterilmelidir.		
Veri, hiçbir işlemten geçmeden çıktığı işleme doğrudan dönemez.		
Veri akış okları üzerinde gösterilen veri, sadece isim formatında olmalıdır.		

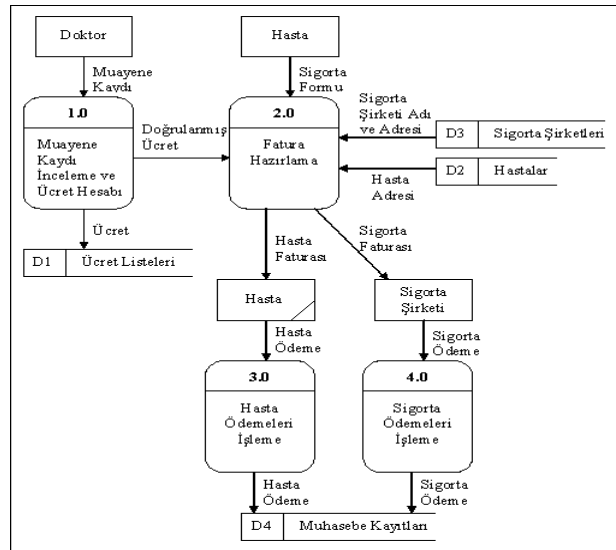
## VAD Düzeyleri: Taslak

- Ön inceleme sonucunda belirlenir
- Sistemle varlıklar arası ilişkiyi gösterir
- Ayrıntılı süreç ve veri depoları bulunmaz
- Bağlam diyagramı olarak da bilinir.



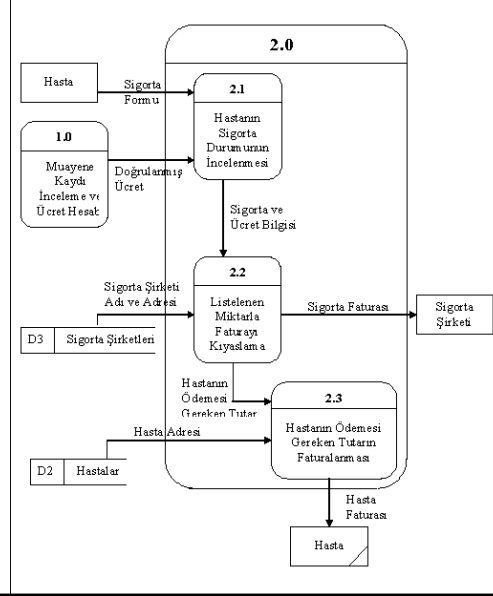
## VAD Düzeyleri: 1.Düzy

- Süreçleri ve işlemleri, ilişkili veri depoları, varlıklar ve depolarla işlemler arası ilişkileri gösterir
- Sistemdeki tüm süreçlerin birbiriyle ve dış kaynaklarla olan ilişkisi belirlenir
- Öncesinde «Bağlam VAD» kullanıldıysa, «Düzy 0 VAD» olarak de isimlendirilir

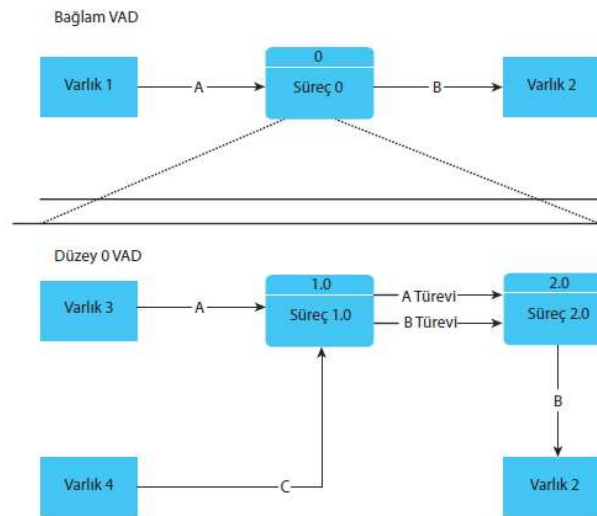


## VAD Düzeyleri: 2.Düzy

- Her sürecin alt işlemleri ayrıntılarıyla gösterilir
- 1.Düzydeki her süreç için bir 2.Düzy VAD çizilir
- Tek bir süreçle veri kaynakları arası ilişki detaylı gösterilir
- Öncesinde «Düzy 0 VAD» kullanıldıysa, «Düzy 1 VAD» olarak de isimlendirilir

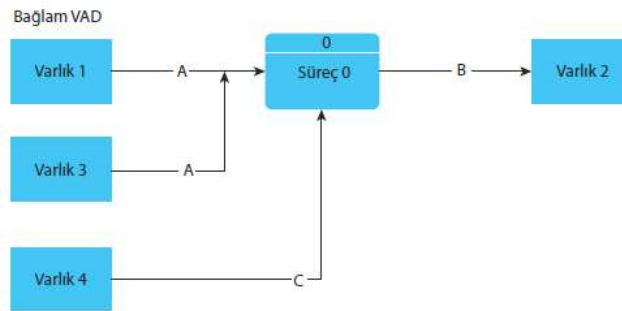


## Hata Nerede?

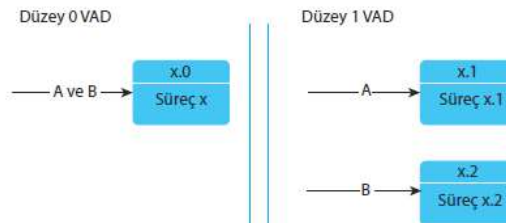


## Çözümü

- Varlık 3 ve Varlık 4 bağlam VAD'da yer almalıdır.



## Burada sorun var mı?





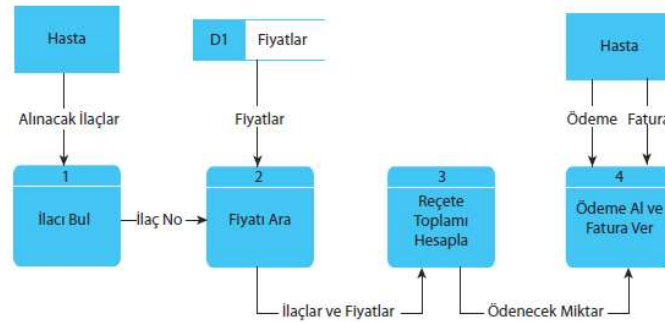
## Kavramsal Veri Modeli - Varlık İlişki Diyagramları

- Yüksek seviyeli, kullanıcı topluluğuyla iletişim için kullanılır.
- Veri yapısı, donanım, yazılım bilgisi yer almaz.
- Tam ve yeterli bileşenler ile sistemin bilgi gereksinimlerinin gerçek gösterimini oluşturur.
- **Varlık ilişki diyagramları:** verinin kavramsal gösterimini sunar.
  - Veri nesneleri ile aralarındaki ilişkilerin grafiksel gösterimi
  - Kavramsal veri modellemesinin yapılmasını sağlar
- İleriki haftalarda

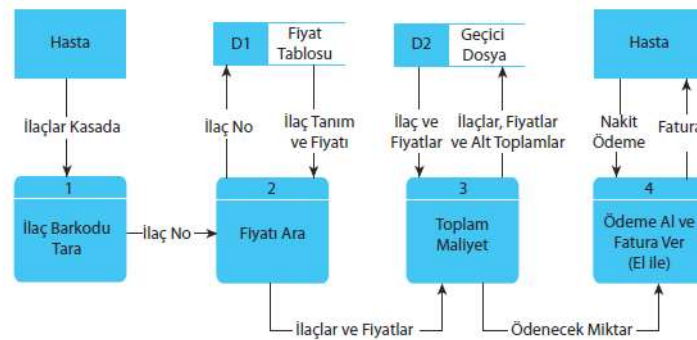
## Mantıksal ve Fiziksel VAD

- Mantıksal VAD: işler nedir?
  - Faaliyetler, varlıklar ve üretilip kullanılacak veriler tanımlanır
  - Analiz safhası
- Fiziksel VAD: sistem nasıl uygulanacak,
  - Hangi donanım, yazılım, dosya, insan kaynağı nasıl kullanılacak?
  - Tasarım safhası

## Hastane Bilgi Sistemi - Eczaneden ilaç temin etme Mantıksal VAD



## Hastane Bilgi Sistemi - Eczaneden ilaç temin etme Fiziksel VAD



## Süreçler Nasıl Modellenecek?

- Süreçler tanımlandı ama içindeki işlemlerle ilgili açıklama yok.
- Mantıksal modellemede süreç iç yapıları ve işlevleri ifade edilir.
- Süreç tanımlama formları kullanılır.
  - Mantık tanımlamak için 3 yöntem vardır:
    - Yapısal dil (sözde kod)
    - Karar tabloları
    - Karar Ağaçları

## Süreç Tanımlama Formu

Süreç Tanımlama Formu	
No	4
Ad	Ödemeyi Al ve Fatura Ver
Tanım	Reçetede ki ilaçların toplam maliyetine bağlı olarak hasta ödemesi nakit ya da kart ile tahsil edilecek ve fatura verilecek.
Girdi Veri Akışı	<ul style="list-style-type: none"><li>• Süreç 3'ten hesaplanarak gelecek olan "ne kadar ödeme yapılacağı" bilgisi,</li><li>• Hastadan yapılacak tahsilat.</li></ul>
Çıktı Veri Akışı	Fatura
Süreç Tipi	<input type="checkbox"/> Çevrimiçi <input type="checkbox"/> Otomatik <input checked="" type="checkbox"/> El İle
Süreç Mantığı	<pre>READ ÖDENECEK_MIKTAR ÖDEME AL SELECT CASE   CASE 1 (ÖDEME büyüktür ÖDENECEK_MIKTAR)     DO Para üstü ver       ÖDEME=Tamam   CASE 2 (ÖDEME eşittir ÖDENECEK_MIKTAR)     DO ÖDEME=Tamam   CASE 3 (ÖDEME küçüktür ÖDENECEK_MIKTAR)     DO ÖDEME=Tamam değil END SELECT IF ÖDEME eşittir Tamam   THEN Fatura ver ELSE   Hastaya bilgi ver END IF</pre>
Çözülmeyen Sorunlar	<ul style="list-style-type: none"><li>• Nakit ödeme dışında bir ödeme biçiminin kabul edilip edilemeyeceği belli değil.</li><li>• Reçete kayıp ya da unutulmuş ise nasıl bir işlem yapılacaktır?</li></ul>

## Yapısal Dil

- Süreçleri mantıksal olarak ifade edebilmek için standart İngilizce kelimelerden oluşan bir alt küme.
- Sıralı (ardışık), karar, durum ve döngü işlemleri için yapı blokları kullanılır.
- Girintili yazı stili kullanılır.

## Yapısal Dil Blokları

Yapısal Dil Bloğu	Örnek
Ardışık Blok Yapısı: Herhangi bir yönlendirmenin bulunmadığı, özel işlemler ya da kontrol gerektirmeyen işlem bloklarıdır.	Eylem #1 Eylem#2 Eylem#3 Eylem#4
Karar İşlem Bloğu: IF ifadesinden sonra verilen koşul doğru ise THEN ifadesinden sonraki eylemler uygulanır; aksi durumda ELSE ifadesinden sonraki eylemler uygulanır.	IF Koşul 1 Doğru THEN Eylem#1 uygula ELSE Eylem#2 uygula END IF
Durum Kontrol Bloğu: Karar işlem bloğunun özel bir tipidir. Bir koşulun izleyebileceği birden fazla durum varsa ve bu durumlardan biri oluştuğunda diğerleri oluşmıyorsa kullanılan blok yapısıdır.	READ Kontrol-edilecek-değer SELECT CASE CASE 1 (Kontrol-edilecek-değer=Koşul1) DO Eylem#1 CASE 2 (Kontrol-edilecek-değer=Koşul2) DO Eylem#2 CASE 3 (Kontrol-edilecek-değer=Koşul3) DO Eylem#3 CASE 4 (Kontrol-edilecek-değer=Koşul4) DO Eylem#4 END CASE
Döngü Blok Yapısı: Koşul sağlanana kadar tekrarlanması gereken eylemler varsa kullanılır.	DO WHILE Koşul1 doğru oldukça Eylem#1 ENDDO Veya DO Eylem#1 UNTIL Koşul1 doğru olunca

## Yapısal Dil Örneği

```
READ İLAÇ_STOK_MIKTARI
SELECT CASE İLAÇ_STOK_MIKTARI
CASE 1 (İLAÇ_STOK_MIKTARI büyükse KRITİK_STOK)
    Herhangi bir eylem uygulama
CASE 2 (İLAÇ_STOK_MIKTARI esitse KRITİK_STOK)
    Eczane çalışanına bildirimde bulun
CASE 3 (İLAÇ_STOK_MIKTARI küçükse KRITİK_STOK)
    Otomatik olarak sipariş üret
CASE 4 (İLAÇ_STOK_MIKTARI esitse SIFIR)
    Muadil ilaçları göster
IF İLAC_URETIMI_DURDU
    THEN İlacın reçeteye yazılmasını engelle
END IF
END CASE
```

## Karar Tabloları

- Karmaşık kararların mantığını belirleme mekanizması
- Durumlar, Kurallar, İşlemler, Kararlardan oluşur

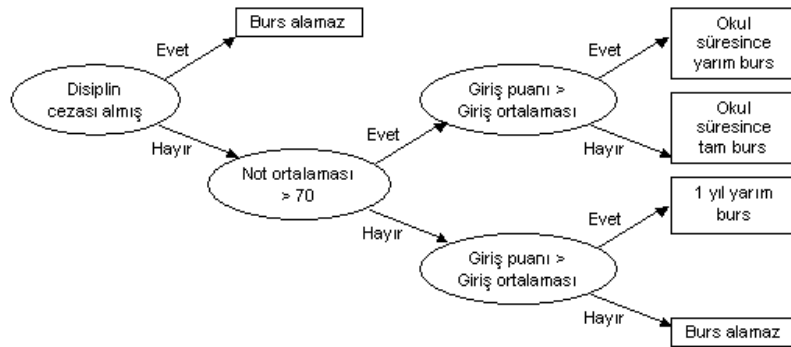
		KURALLAR				
		1	2	3	4	5
DURUM	Disiplin cezası almış	E	H	H	H	H
	Not ortalaması > 70	H	E	H	E	H
	Giriş puanı > Giriş ortalaması	H	E	E	H	H
İŞLEM	Okul süresince tam burs		X			
	1 yıl yarım burs			X		
	Okul süresince yarım burs				X	
	Burs alamaz	X				X

## Karar Tablosu Örneği

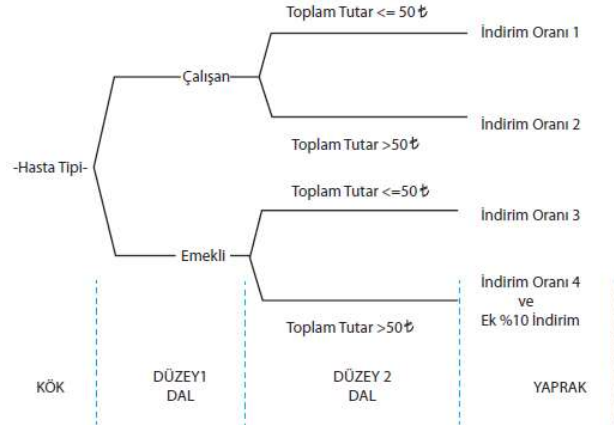
İlaç İndirim Hesapla	KURALLAR			
KOŞULLAR	1	2	3	4
Çalışan Hasta	E	E	H	H
Emekli Hasta	H	H	E	E
Toplam Tutar 50 TL'den küçük ya da eşit	E	H	E	H
Toplam Tutar 50 TL'den büyük	H	E	H	E
EYLEMLER				
İndirim Oranı 1 Uygula	X			
İndirim Oranı 2 Uygula		X		
İndirim Oranı 3 Uygula			X	
İndirim Oranı 4 Uygula				X
Ek %10 İndirim Uygula				X

## Karar Ağaçları

- Daha az karmaşık yapılar için uygun
- Olasılıklar kullanılmaz



## Karar Ağacı Örneği



## Veri Sözlüğü

- Sadece VAD yeterli değil
- Tüm bilişim maddeleri tanımlanmalı
  - Veri Akış Sözlük Girişi
  - Veri Deposu Sözlük Girişi
  - Veri Yapısı Sözlük Girişi
  - Veri Elemanı Sözlük Girişi
  - İşlem Sözlük Girişi
- Veri akış ve işlemler (kısa tanımlanmışlardı) sözlükte tanımlanmalı
- Bileşik veriler öğelerine göre, basit veriler anlamlarına göre tanımlanır

## Veri Akış Sözlük Girişi

### Veri Akış Sözlük Girişi

<b>Veri Akış Adı</b>	: FATURA
<b>Tanım</b>	: Müşteriye fatura edilecek doküman için gerekli bilgiler
<b>Nereden</b>	: 1.1 Faturayı Hazırla
<b>Nereye</b>	: 1.2 Fatura numarasını hazırla
<b>Veri Yapıları</b>	: Fatura Detayları (K) Müşteri Detayları (K) (K: Kompozit, E: Elemanter)
<b>Açıklama</b>	: .....

## Veri Deposu Sözlük Girişi

### Veri Deposu Sözlük Girişi

<b>Veri Depo Adı</b>	: SATIŞ SİPARİŞ FORM DOSYASI
<b>Tanım</b>	: Satış sipariş formlarının saklandığı arşiv dosyasıdır
<b>Veri Yapıları</b>	: Satış sipariş kaydı
<b>Miktar</b>	: Günde yaklaşık 100 kayıt
<b>Erişim</b>	: Sipariş bölümü personeli
<b>Açıklama</b>	: .....



## Veri Yapısı Sözlük Girişi

### Veri Yapısı Sözlük Girişi

(Her kompozit veri yapısı için olmalıdır)

<b>Veri Yapı Adı</b>	: SATIŞ SİPARİŞ KAYDI
<b>Tanım</b>	: Müşterinin mal siparişi için kullandığı satış sipariş formu
<b>Veri Elemanları</b>	: MusteriNo (E) SiparisNo (E) SiparisTarihi (K) * ParcaNo (E) * Miktar (E) * BirimFiyat (E)
<b>Açıklama</b>	: * olanlar, her bir parça kaydı için oluşur.

## Veri Elemanı Sözlük Girişi

### Veri Elemanı Sözlük Girişi

<b>Veri Elemanı Adı</b>	: MusteriNo
<b>Tanım</b>	: Müşteriyi tanımlayan numara
<b>Tip</b>	: Numerik
<b>Uzunluk</b>	: 4
<b>Değer Aralığı</b>	: 0001-6999
<b>Diğer Detaylar</b>	: .....
Her elemanter veri yapısı için. Sadece değer ise kod tablosu da olabilir	

## İşlem Sözlük Girişi

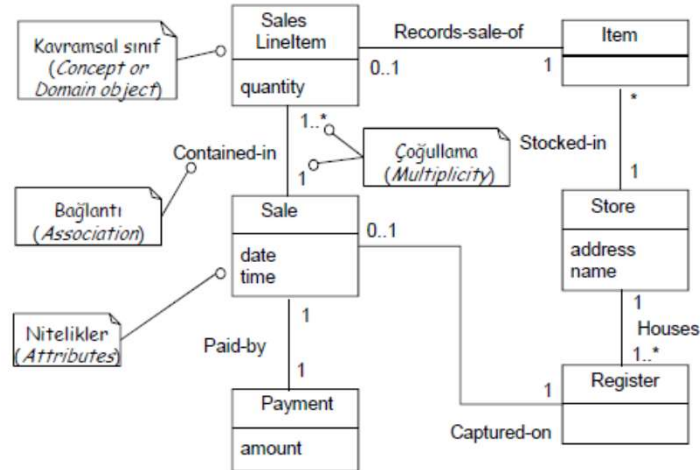
### İşlem Sözlük Girişi

**İşlem Adı** : 2.0 Sipariş Satış Verisini Gir  
**Girdi** : Satış siparişleri  
**İşlem Tanımı** : ... yap, eğer değilse ..... yap vb.  
**Çıktı** : Girilmiş satış siparişleri

## Problem Uzayının Modellenmesi

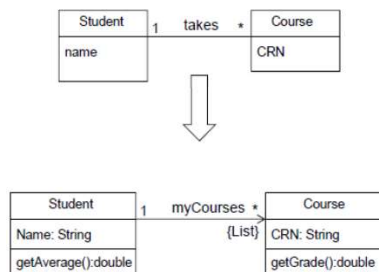
- Amaç gerçek dünyanın (problemin) doğru, anlaşılır, sınanabilir modelini oluşturmak
- Problemin anlaşılması aşamasıdır
- Teknik Yorumdan çok müşteri isterleri ön plandadır
- İsteklerin modellenmesi nd değil, burası nd.
- Gerçek dünyadaki kavramsal sınıflar ve nesneler yer alır (nesneler, o anda görülen özellikleri, aralarındaki ilişkiler (bağlantılar)). Yazılım nesneleri henüz düşünülmez
- UML ile görselleştirilir.
- Oluşacak sistem anlaşılır.
- Tasarıma girdi sağlar (sorumlulukları atamak üzere)

## Kavramsal Sınıf Diyagramı



## Problem ile Yazılımın Yakınlaşması

- Tasarımda yazılım sınıfları oluşturulurken ve sorumluluklar atanırken uygulama modeli kullanılır



## Kavramsal Sınıfların Bulunması

- Gerçek dünyadaki somut/soyut varlıklar
- **Kategori listesi:** Deneyim sonucu sık karşılaşılan kategoriler. Zaten görülen sınıfları bulur. Gözden kaçırma olasılığı yüksek.
- **Senaryolardaki isimler (isim tamlamaları):** Senaryolardaki tüm isim ve tamlamalar aday sınıf olarak alınır. Çoğunlukla fazla sınıf çıkar, elenir.
- **Varolan model güncellemesi:** Yayımlanmış modellerin uyarlaması yapılır.

## Örnek: İsim İşaretleme

### Main Success Scenario (or Basic Flow):

1. **Customer** arrives at a **POS checkout** with **goods** and/or **services** to purchase.
  2. **Cashier** starts a new **sale**.
  3. Cashier enters **item identifier**.
  4. System records **sale line item** and presents **item description, price**, and running **total**. Price calculated from a set of price rules.
- Cashier repeats steps 3-4 until indicates done.*
5. System presents total with **taxes** calculated.
  6. Cashier tells Customer the total, and asks for **payment**.
  7. Customer pays and System handles payment.
  8. System logs completed sale and sends sale and payment information to the external **Accounting** system (for accounting and **commissions**) and **Inventory** system (to update inventory).
  9. System presents **receipt**.
  10. Customer leaves with receipt and goods (if any).

### Extensions:

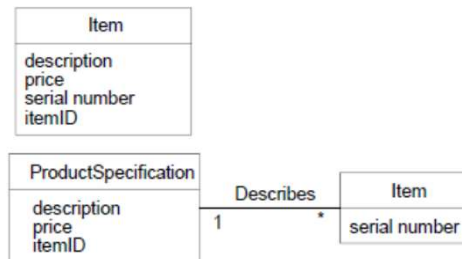
- 7a. Paying by cash:
1. Cashier enters the cash **amount tendered**.
  2. System presents the **balance due**.
- .....

## Gereksiz Sınıfların Elenmesi

- **Fazlalık sınıflar:** aynı unsuru ifade edenlerden, daha tanımlayıcı olan. Kişi-müşteri: Müşteri
- **İlgisiz sınıflar:** çözümle ilgisi olmayan ya da o aşamada ilgilenmeyeceğimiz sınıflar: Kredi Kartı
- **Belirsiz sınıflar:** sınırları iyi çizilmemiş, kaba-geniş tanımlı sınıflar. Genellikle birden fazla sınıftan oluşmuşlardır ya da başka sınıfın parçasıdır: Muhasebe Sistemi
- **Nitelikler:** kendi başına varlığı anlamlı olmayanlar, sınıfların özellikleridir: Miktar
- **İşlemler:** Sadece nesnelere uygulanan işlemler sınıf olmaz. Kendi nitelikleri olmalı ve hizmet alışverişi yapmalıdır: Ödeme(miktar, para birimi, tarih, ...)
- **Roller:** Sınıflar arası ilişki olan roller sınıf olamaz.
- **Gerçekleme Elemanları:** Yazılım sınıfları bu uzayda yer almaz.
- **Sınıf olup olmadığının testi için şu sorular sorulabilir:**
  - Kavramla ilgili veri saklanması gerekiyor mu?
  - Değişik değerler alabilecek farklı özellikleri var mı?
  - Kavramdan birçok nesne türeyebilir mi?
  - Uygulamanın kapsama alanı içinde mi?
  - Sınırları iyi çizilmiş mi, tanımı yapılabiliyor mu?

## Betimleme Sınıfı (description-specification) İhtiyacı

- Satış yaparken nesneler ürünleri tutsun.
- Bir cinsten tüm malzeme satılırsa, nesne kalmaz. Bilgi kaybı olur.
- Bu yüzden özellikleri ayrı bir sınıf olarak tutmak gerekebilir.



## Bağlantıların Belirlenmesi

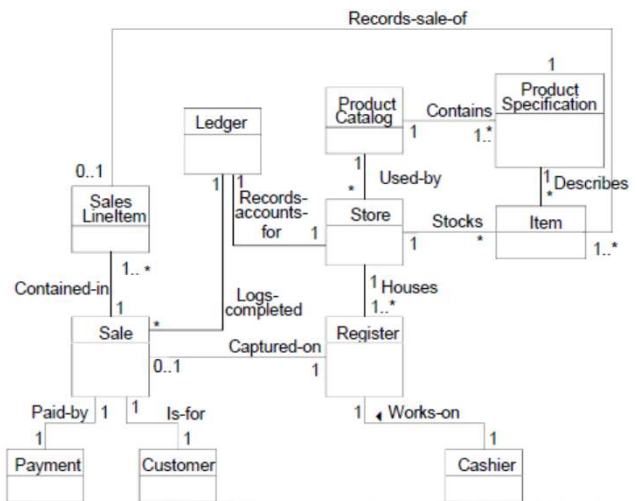
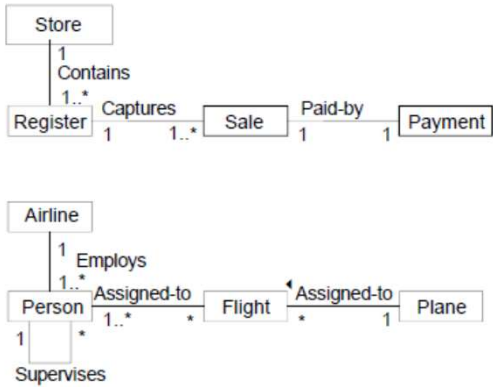
- Bir nesnenin kaç tane nesneyle ilişki içinde olacağı çoğullama ile gösterilir.
- Bağlantılara doğru isimler verilmelidir. Tip-fiil-tip
- Bazı bağlantıların unutulması tasarımı çok etkilemez.
- Kavramsal sınıfların doğru bulunması daha önemlidir.
- Fazla bağlantı anlaşılabilirliği azaltır.
- Yaygın bağlantılar listesinden yararlanılabilir:
  - Fiziksel barındırma, mantıksal barındırma, kayıt ilişkisi, kullanım ilişkisi, tanım, sahiplik, ...
  - Kullanım senaryolarındaki fiillerden yararlanılabilir.

## Gereksiz Bağlantıların Elenmesi

- Elenen sınıflar arası bağlantılar gereksizdir
- Sistemin amacı dışındaki bağlantılar gereksizdir
- Gerçeklemeyi ilgilendiren bağlantılar gereksizdir
- Faaliyetler bağlantı değil, etkileşimdir. (ATM kredi kartı kabul eder)
- Üçlü bağlantılar ikili hale çevrilmelidir. Memur hesapla ilgili işlemleri girer → memur işlemleri girer. İşlemler hesapla ilgilidir.

## Örnek: Bağlantılar

- Tasarımı ilgilendiren bağlantıları ortaya çıkarın
- Tip-fiil-tip doğru isim ataması yapın.



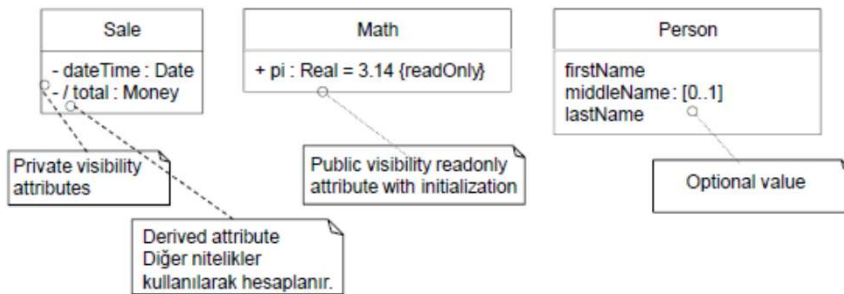
## Özelliklerin Belirlenmesi

- Özellik: nesne yaratıldığında nesneye özgü değer alabilen veri.
- Senaryolarla ilgili nitelikler bulunmalıdır.
- Basit veri tipleri ile ifade edilirler.
  - birimleri varsa doğru değil. (para gibi)
- Karmaşık tipteyse, başka bir sınıf olma olasılığı yüksektir.
  - birden fazla alansa (tel no, ad soyad)
  - işlem yapılıyorsa (kredi kartı ve doğrulama)
  - kendi nitelikleri varsa (fiyat: geçerlilik tarihi))

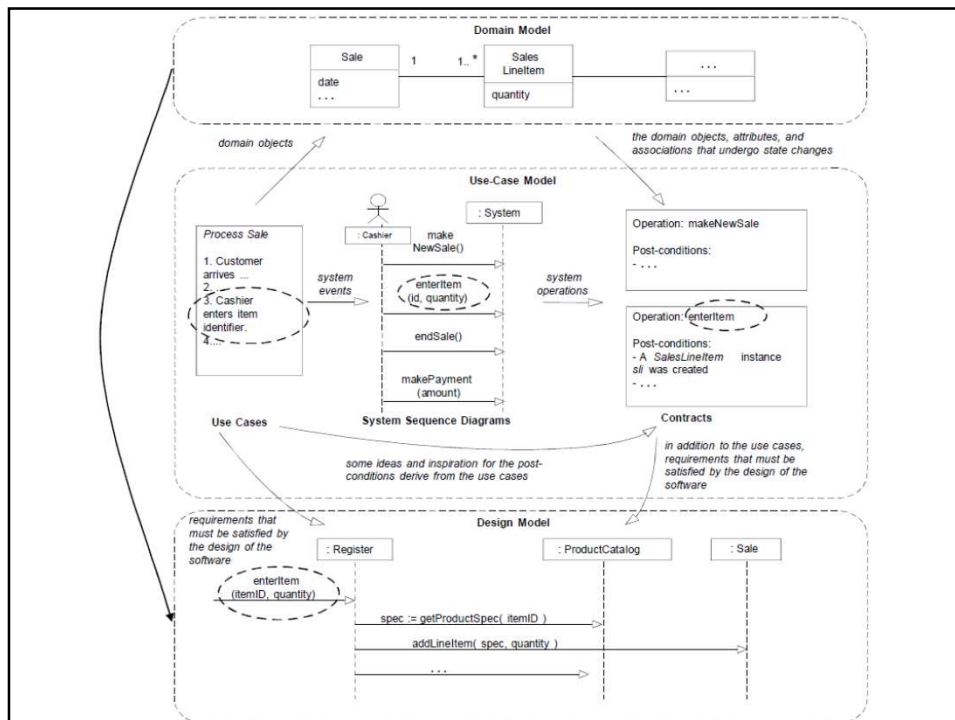
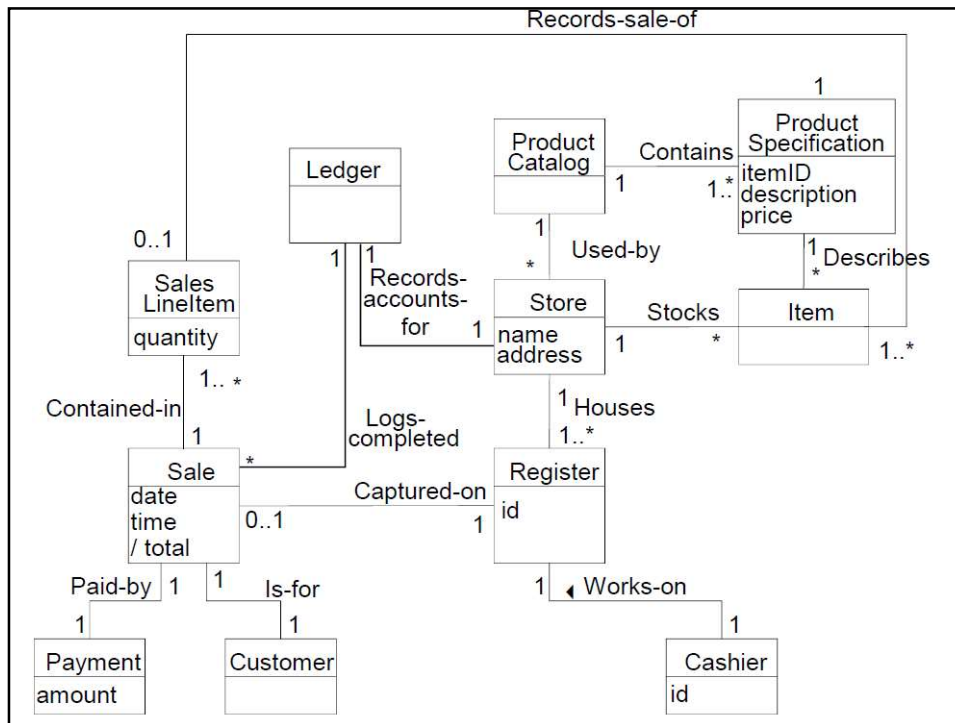


## Özellikler-Detay

- Analizde kavramsal sınıf özellikleri hakkında detay bilgi varsa, tasarıma kaynak olması için bunlar da belirtilebilir.







## Gelecek Dersler

- Sistem Analizi Laboratuvar Uygulaması (VAD+Kavramsal Sınıf D.)
- Sistem Tasarımı
  - Girdi, Çıktı, Veri Yapısı, Arabirim Tasarımları
- Ara Sınav