



ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
**UZAKTAN EĞİTİM UYGULAMA VE
ARAŞTIRMA MERKEZİ**

VERİ TABANI YÖNETİM SİSTEMLERİ

İÇİNDEKİLER

❖TÜMLEŞİK MODELLEME DİLİ (Unified Modeling Language - UML)

UML NEDİR?

- Yazılım ve donanımların bir arada düşünülmesi gereken,
- Zor ve karmaşık programların,
- Özellikle birden fazla yazılımcı tarafından kodlanacağı durumlarda,
- Standart sağlamak amacıyla endüstriyel olarak geliştirilmiş grafiksel bir dildir.
- ~~Programlama dili~~ Diyagram çizme ve ilişkisel modelleme dili

UML NEDİR?

- UML'in doğuşu son yıllarda yazılım endüstrisindeki en büyük gelişmelerden biri olarak kabul edilebilir.
- UML 1997 yılında yazılımın, diyagram şeklinde ifade edilmesi için bir standartlar komitesi tarafından oluşturuldu.
- Diğer dallardaki mühendislerin standart bir diyagram çizme aracı -Programcıların UML

UML NEDİR?

- UML; yazılım sisteminin önemli bileşenlerini tanımlamayı, tasarlamayı ve dokümantasyonunu sağlar.
- Yazılım geliştirme sürecindeki tüm katılımcıların (kullanıcı, iş çözümleyici, sistem çözümleyici, tasarımcı, programcı,...) gözüyle modellenmesine olanak sağlar.
- UML gösterimi nesneye dayalı yazılım mühendisliğine dayanır.

UML Faydası

- Yazılımın geniş bir analizi ve tasarımı yapılmış olacağından kodlama işlemi daha kolay ve doğru olur.
- Hataların en aza inmesine yardımcı olur.
- Geliştirme ekibi arasındaki iletişimi kolaylaştırır.
- Tekrar kullanılabilir kod sayısını artırır.
- Tüm tasarım kararları kod yazmadan verilir.
- Yazılım geliştirme sürecinin tamamını kapsar.
- “resmin tamamını” görmeyi sağlar.

Grafiksel Gösterimler

UML Grafiksel Gösterimler

Yapısal Diyagramlar

Davranışsal Diyagramlar

Sınıf

Nesne

Bileşen

Dağılım

Kullanım
Senaryosu

Ardışık

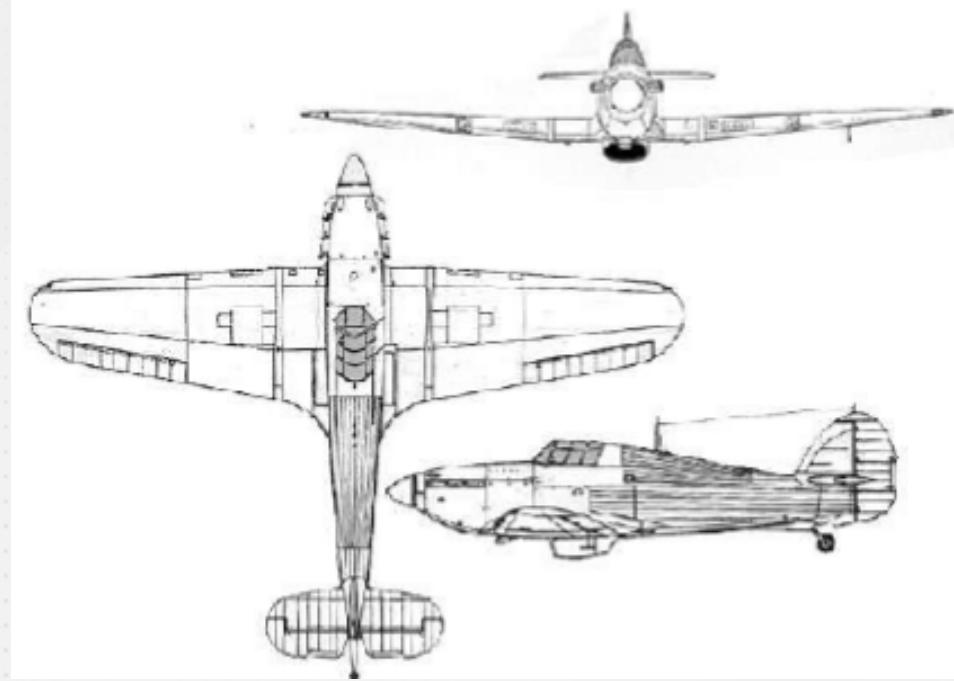
İşbirliği

Durum

Etkinlik

Grafiksel Gösterimler

- Bu grafiksel gösterimler aynı şeye farklı şekillerde bakabilmeyi sağlar.



https://www.the-blueprints.com/blueprints/ww2planes/ww2-hawker/36486/view/hawker_hurricane_1935/

4+1 bakış

- Bir yazılım sistemi oluşturulurken sadece tek boyutta analiz ve modelleme yapılmaz.
- Bu amaçla; yazılım geliştirme sürecinin farklı aşamalarında farklı UML diyagramlarını kullanmak gerekmektedir.
- 4+1 bakış, bu diyagramları sınıflandırmak ve yazılım yaşam çevrimindeki kullanım yerlerini ortaya koymak için kullanılan bir kavramdır.

4+1 bakış

- 1 • Kullanıcı Bakış (User View)
- 2 • Yapısal Bakış (Structural View)
- 3 • Davranış Bakış (Behavioral View)
- 4 • Gerçekleme Bakış (Implementation View)
- 5 • Ortam Bakış (Environment View)

4+1 bakış

- 1 • Kullanıcı Bakışı (User View)
- 2 • Yapısal Bakış (Structural View)
- 3 • Davranış Bakışı (Behavioral View)
- 4 • Gerçekleme Bakışı (Implementation View)
- 5 • Ortam Bakışı (Environment View)

4+1 bakış

1-) Kullanıcı Bakışı (User View)

- Müşteri gereksinimlerini ortaya koymak ve müşteriye, sistemi tanıtmak amacı ile kullanılan bakış açısıdır.
- Bazı kaynaklarda; ***kullanım senaryosu (use case)*** bakışı olarak da açıklanmaktadır.
- Bu amaçla "kullanım senaryosu" (use case) diyagramları kullanılmaktadır.

4+1 bakış

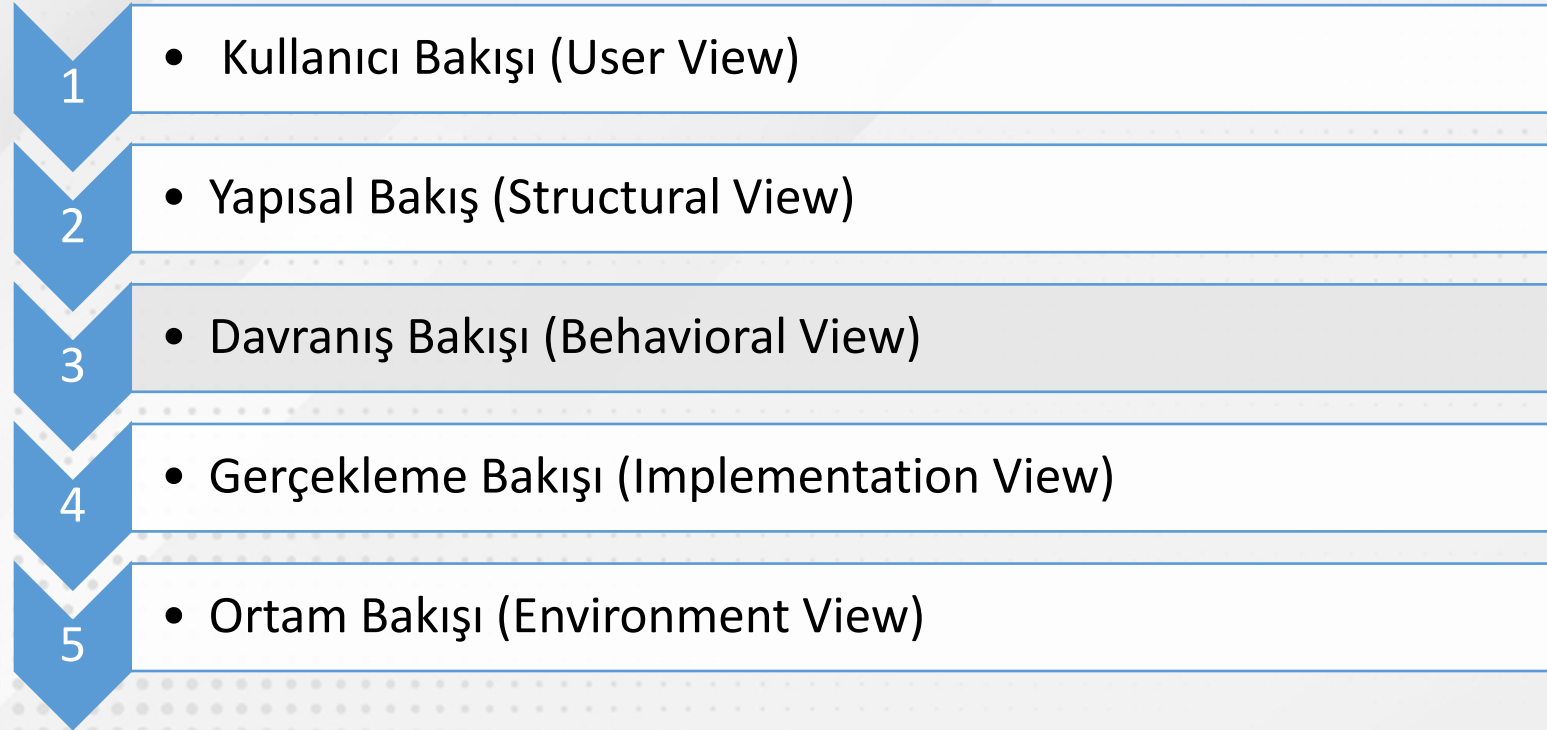
- 1 • Kullanıcı Bakışı (User View)
- 2 • Yapısal Bakış (Structural View)
- 3 • Davranış Bakışı (Behavioral View)
- 4 • Gerçekleme Bakışı (Implementation View)
- 5 • Ortam Bakışı (Environment View)

4+1 bakış

2-) Yapısal Bakış (Structural View)

- Sistemin nelerden meydana geldiğini gösteren bakış açısıdır.
- Bu amaçla; "sınıf" (class) diyagramları ve "nesne" (object) diyagramları kullanılmaktadır.

4+1 bakış



4+1 bakış

3-) Davranış Bakışı (Behavioral View)

- Sistemin dinamik yapısını ortaya koyan bakış açısıdır.
- Bu amaçla; «*ardışık*» (*sequence*), "*işbirliği*" (*collaboration*), "*durum*« (*state*) ve «*etkinlik*" (*activity*) diyagramları kullanılmaktadır.

UML: İlk Bakış

- **Sıralı Diyagramlar (Sequence diagrams)**
Sistemin objeler arasındaki dinamik davranışını tarif eder.
- **İşbirliği Diyagramları (Collaboration diagrams)**
Nesneler arasındaki ilişkileri ve aralarında gezen mesajların sırasını gösterir.
- **Durum Diyagramları (State chart diagrams)**
Tek bir objenin dinamik davranışını tarif eder.
- **Aktivite Diyagramları (Activity diagrams)**
Sistemin dinamik davranışını tarif eder, özelde iş akışını tarif eder.

4+1 bakış

- 1 • Kullanıcı Bakışı (User View)
- 2 • Yapısal Bakış (Structural View)
- 3 • Davranış Bakışı (Behavioral View)
- 4 • Gerçekleme Bakışı (Implementation View)
- 5 • Ortam Bakışı (Environment View)

4+1 bakış

4-) Gerçekleme Bakışı (Implementation View)

- Sistemin alt modüllerini ortaya koyan bakış açısıdır.
- Bu amaçla; "bileşen" (component) diyagramları kullanılmaktadır.

4+1 bakış

- 1 • Kullanıcı Bakışı (User View)
- 2 • Yapısal Bakış (Structural View)
- 3 • Davranış Bakışı (Behavioral View)
- 4 • Gerçekleme Bakışı (Implementation View)
- 5 • Ortam Bakışı (Environment View)

4+1 bakış

5-) Ortam Bakışı (Environment View)

- Donanımın, fiziksel mimarisinin ortaya konduğu bakış açısıdır.
- Bu amaçla; "dağıtım" (deployment) diyagramları kullanılmaktadır.

Sınıf Diyagramları

- Sınıf Diyagramları UML 'in en sık kullanılan diyagram türüdür.
- Sınıflar, nesne tabanlı programlama mantığından yola çıkarak tasarlanmıştır.
- Sınıf diyagramları bir sistem içerisindeki nesne tiplerini ve birbirleri ile olan ilişkileri tanımlamak için kullanılırlar.

Sınıf Diyagramları

- Sınıfların
 - bir adı
 - nitelikleri ve
 - işlevleri vardır
 - Bunlara ek olarak
 - “notes”
 - “Constraints”

SınıfAdı
Özellik 1 : tür 1 Özellik 2 : yaş="19" ...
İşlev 1() İşlev 2(parametreler) İşlev 3():geri dönen değer tipi ...

Sınıf Diyagramları

- Örnek:

SınıfAdı
Özellik 1 : tür 1 Özellik 2 : yaş="19" ...
İşlev 1() İşlev 2(parametreler) İşlev 3():geri dönen değer tipi ...

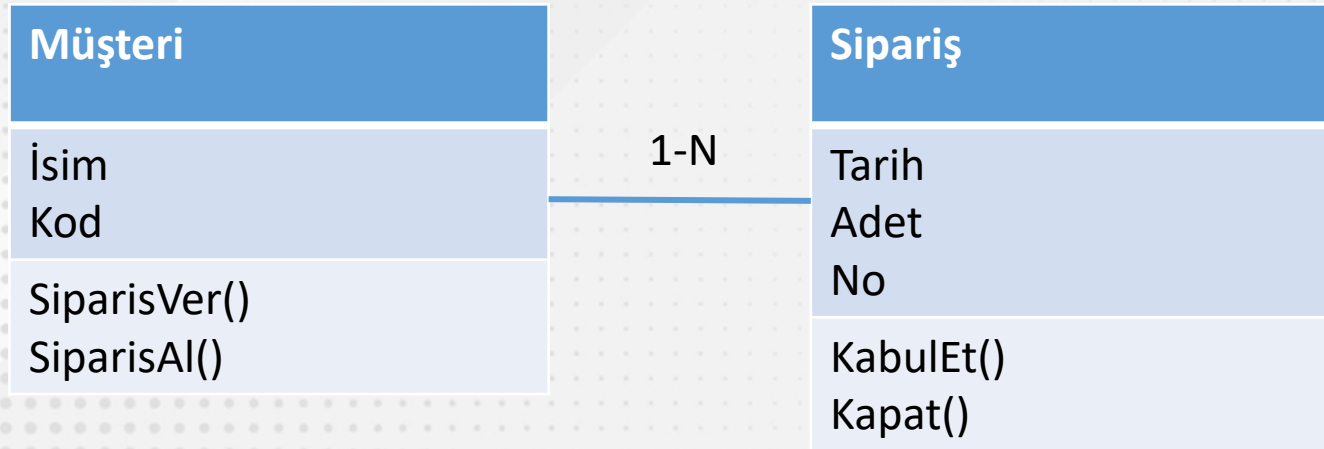
Müşteri

İsim
Kod

SiparisVer()
SiparisAl()

Sınıf Diyagramları

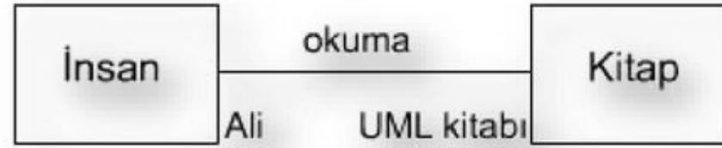
- Sınıflar arasındaki ilişkiyi göstermek için iki sınıf arasına düz bir çizgi çekilir.
- İlişkiyi gösteren çizginin üzerine ilişkinin türü yazılır.



Sınıf Diyagramları

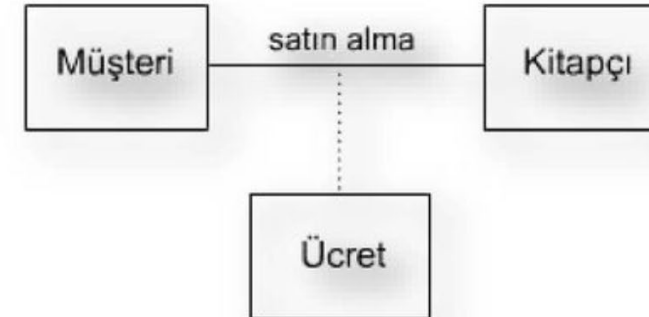
- Bazı durumlarda sınıflar arasındaki ilişki, bir çizgiyle belirtebilecek şekilde basit olmayabilir.
- Bu durumda ilişki sınıfları kullanılır.
- İlişki sınıfları bildigimiz sınıflarla aynıdır.
- Sınıflar arasındaki ilişki eğer bir sınıf türüyle belirleniyorsa UML ile gösterilmesi gerekir.

Sınıflar Arasındaki İlişkiler

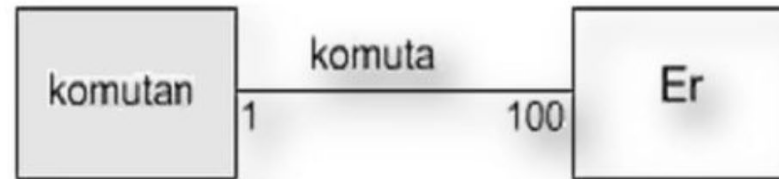


Burada müşteri ile kitapçı sınıfı arasında "satın alma" ilişkisi var. Fakat müşteri satın alırken ücret ödemek zorundadır. Bu ilişkiyi göstermek için ücret sınıfı ilişki ile kesikli çizgi ile birleştirilir.

İnsan sınıfından Ali nesnesi ve Kitap sınıfından 'Uml Kitabı' nesnesi ve aralarındaki 'okuma' ilişkisi



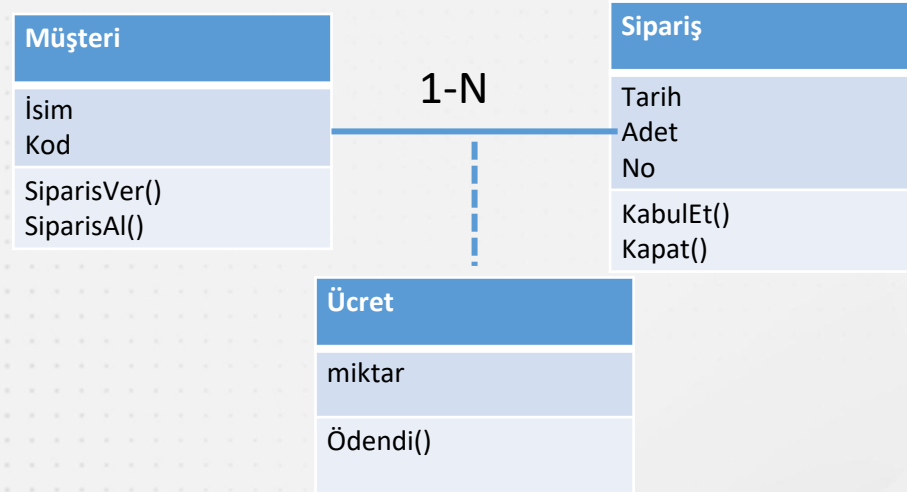
Burada 1 komutan 100 Er'e komut(emir) verebilir anlamı çıkmaktadır



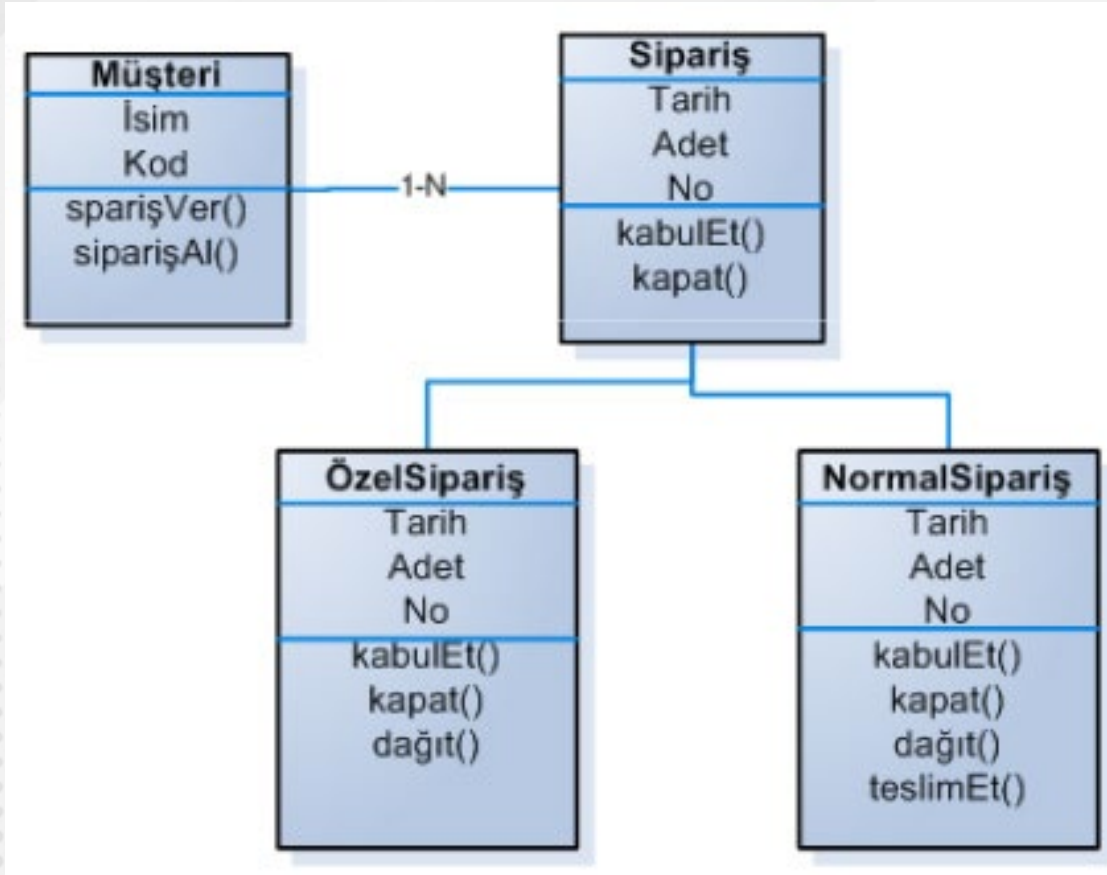
<https://slideplayer.biz.tr/slide/12836589/>

Sınıf Diyagramları

- Müşteri ile Sipariş sınıfı arasında ilişki vardır. Fakat müşteri satın alırken Ücret ödemek zorundadır
- Bu ilişkiyi göstermek için Ücret sınıfı ilişki ile kesikli çizgi ile birleştirilir.



Örnek Sınıf Diyagramı

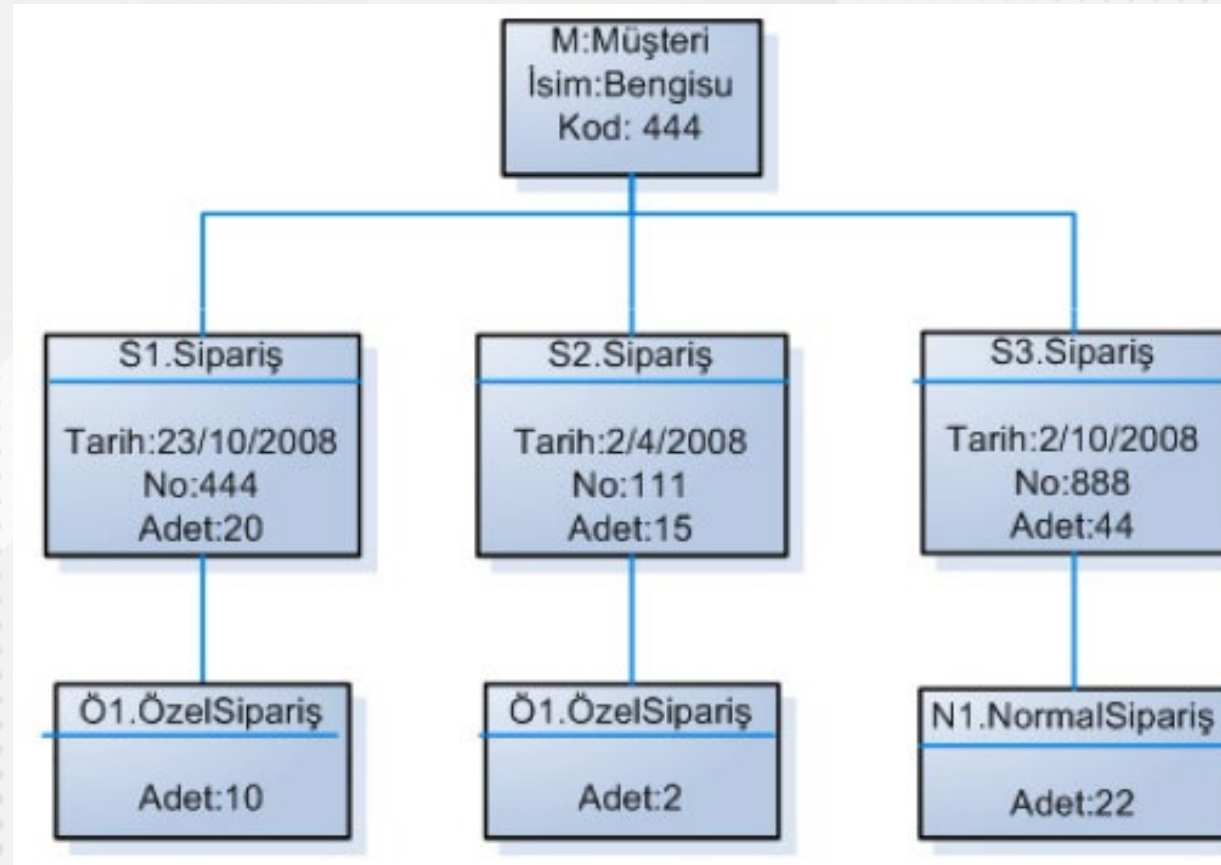


<https://docplayer.biz.tr/105386431-Tumlesik-modelleme-dili-uml-unified-modeling-language.html>

Nesne Diyagramları

- Nesne Diyagramları nesneler ve aralarındaki bağıntıları gösterirler.
- Nesneler, sınıfların somut örnekleri olduğundan sınıf özelliklerinin değeri almış halidirler. Yani, sınıf diyagramlarından türetilir.

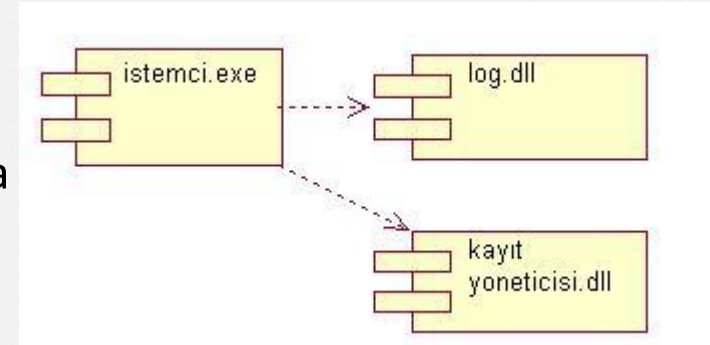
Örnek Nesne Diyagramı



<https://docplayer.biz.tr/105386431-Tumlesik-modelleme-dili-uml-unified-modeling-language.html>

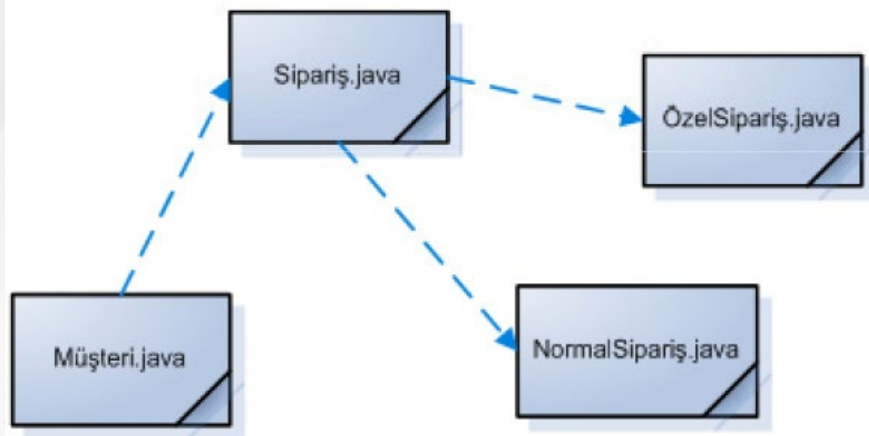
Bileşen Diyagramları

- Bileşen Diyagramları yazılım sistemine daha yüksek bir seviyeden bileşenler seviyesinden bakabilmeyi sağlar.
- Bunlar sistemdeki sınıflar, çalıştırılabilen program parçaları, kütüphane bileşenleri veya veritabanı tabloları olabilir.
- Bu gösterim bileşenler, arayüzler ve bağımlılık ilişkilerinden oluşur ve sistemin fiziksel gösterimini sağlar.

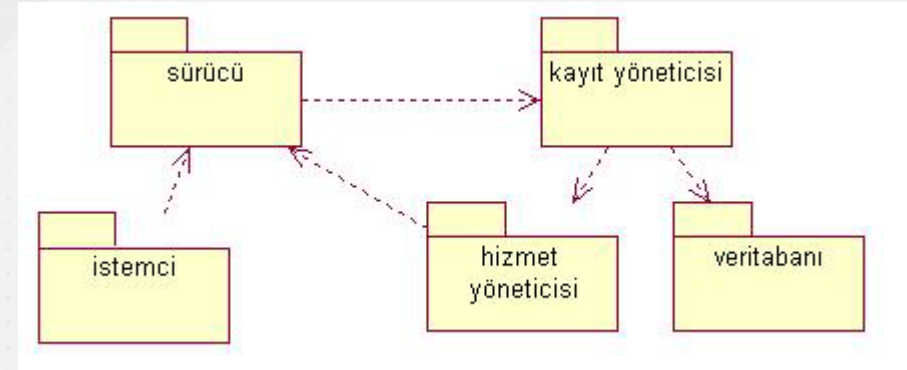


<http://www.csharpnedit.com/articles/read/7?id=482&title=UML%20Bileşen%20%28Component%29%20Diyagramları>

Örnek Bileşen Diyagramı



<https://docplayer.biz.tr/105386431-Tumlesik-modelleme-dili-uml-unified-modeling-language.html>

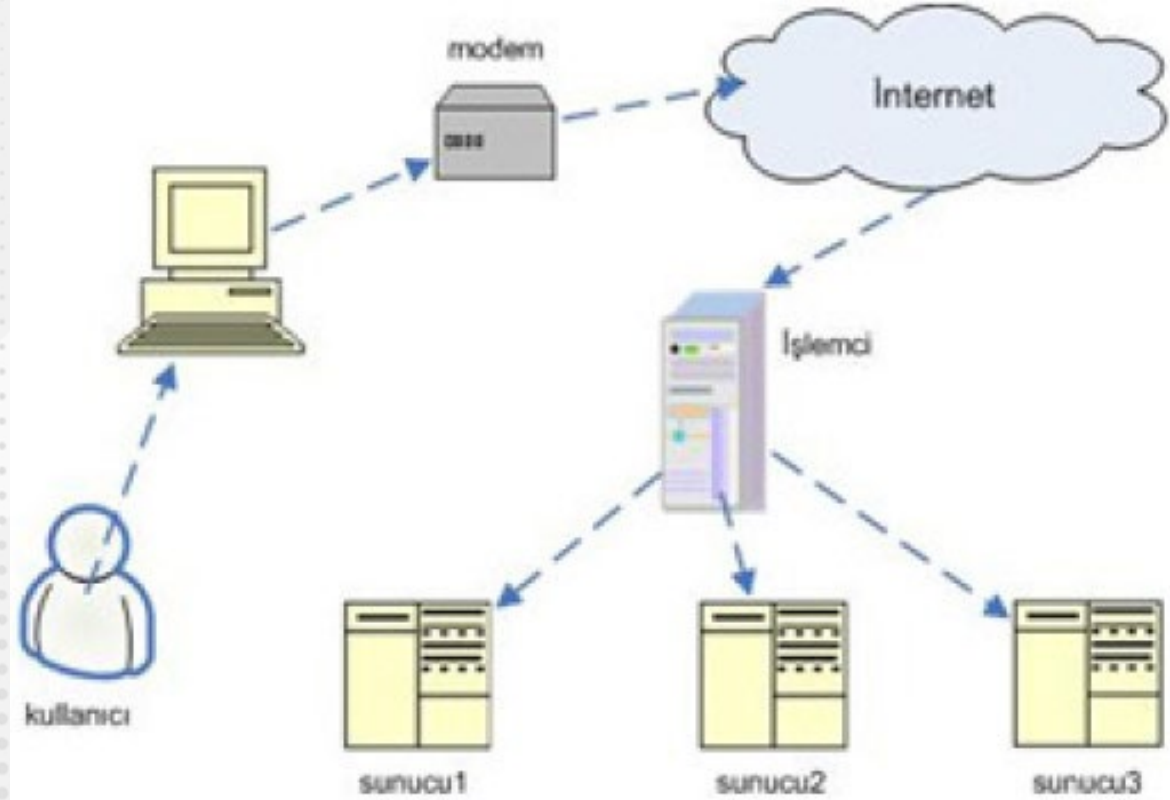


<http://www.csharpnedir.com/articles/read/?id=482&title=UML%20ile%20Bileşen%20Component%29%20Diyagramları>

Dağılım Diyagramları

- Sistemin çalışma platformundaki durumlarını gösterirler.
- Yapısal diyagramdır ve modellenen sisteme belli bir açıdan bakışı gösterir.
- Sistemdeki yürütülebilen parçalar, kütüphaneler, tablolar ve dosyalar gibi bileşenlerin dağılımını belirler.
- Bu diyagramlar, sistem donanım mimarisinin gösterilmesi, gerekli donanım bileşenlerinin tanımlanması amacıyla kullanılabilirler.

Örnek Dağılım Diyagramı



<https://gtumcain.wordpress.com/2011/11/09/uml-deployment-diagram/>

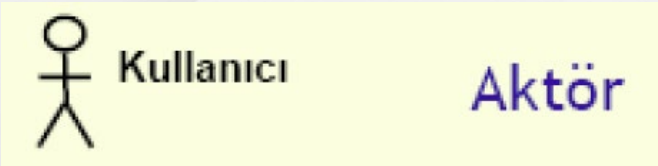
Kullanım Senaryosu Diyagramları

- Kullanım senaryosu diyagramları sistemin işlevsel gereksinimlerinin ortaya çıkarılması için kullanılır.
- Sistemin kullanıcısının bakış açısıyla modellenmesi amacıyla kullanılır.
- Kullanım senaryosu diyagramları sistemin kabaca ne yaptığı ile ilgilenir, kesinlikle nasıl ve neden yapıldığını incelemmez.

Kullanım Senaryosu Diyagramları

- **Kullanım senaryosu**
 - aktörler (sistem dışı aktörler –kullanıcılar veya diğer sistemler)
 - senaryolar (kullanıcı tarafından başlatılan çeşitli olaylar dizisi)

Kullanım Senaryosu Diyagramları

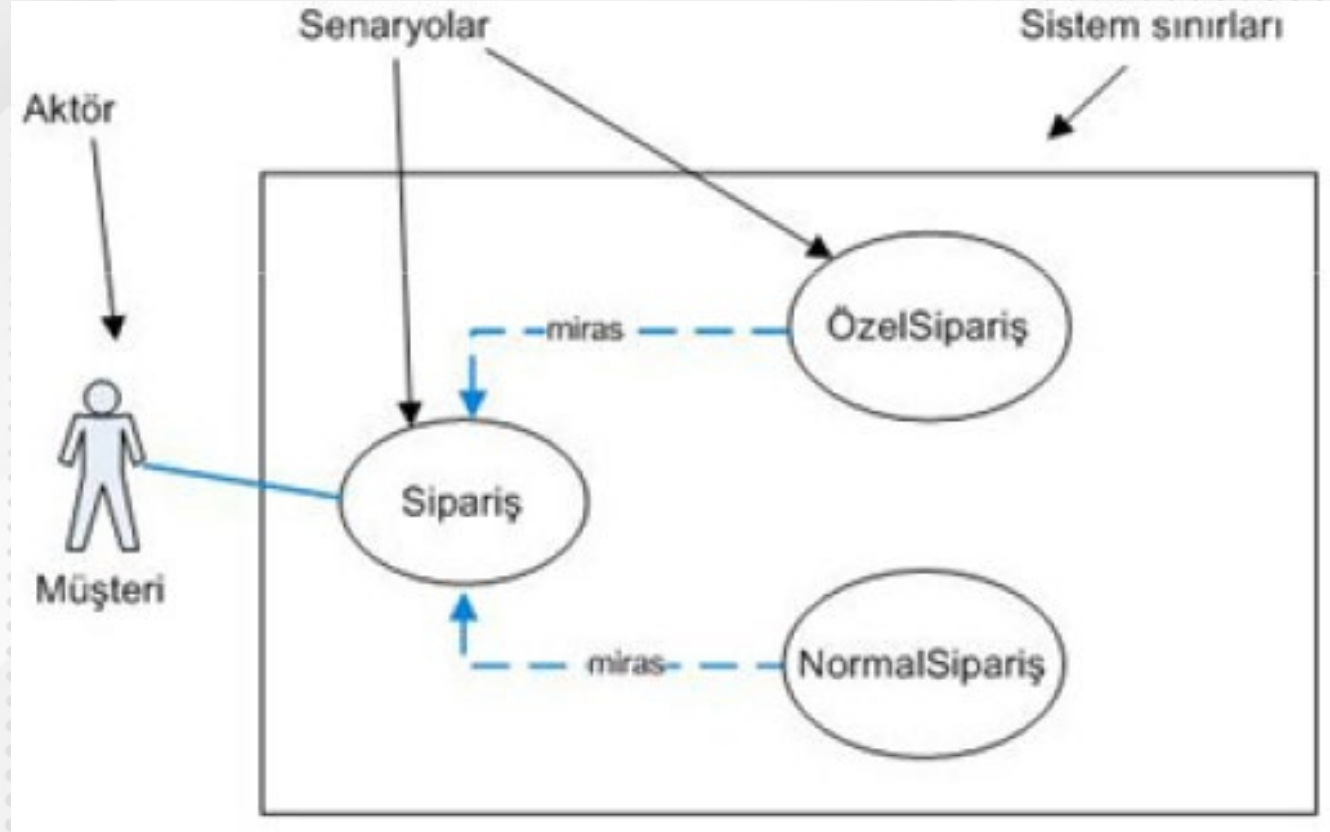


- * Aktör genellikle “insan” olarak düşünölmekle birlikte başka “sistem” ve “donanım” da olabilir.
- * Aktör sistemi “uyarır”, işlevleri haricen “tetikler”(aktif) yada sistemden “uyarıcı alır”(pasif).
- * Aktör sistemin parçası değildir, “harici” dir.

Kullanım Senaryosu Diyagramları

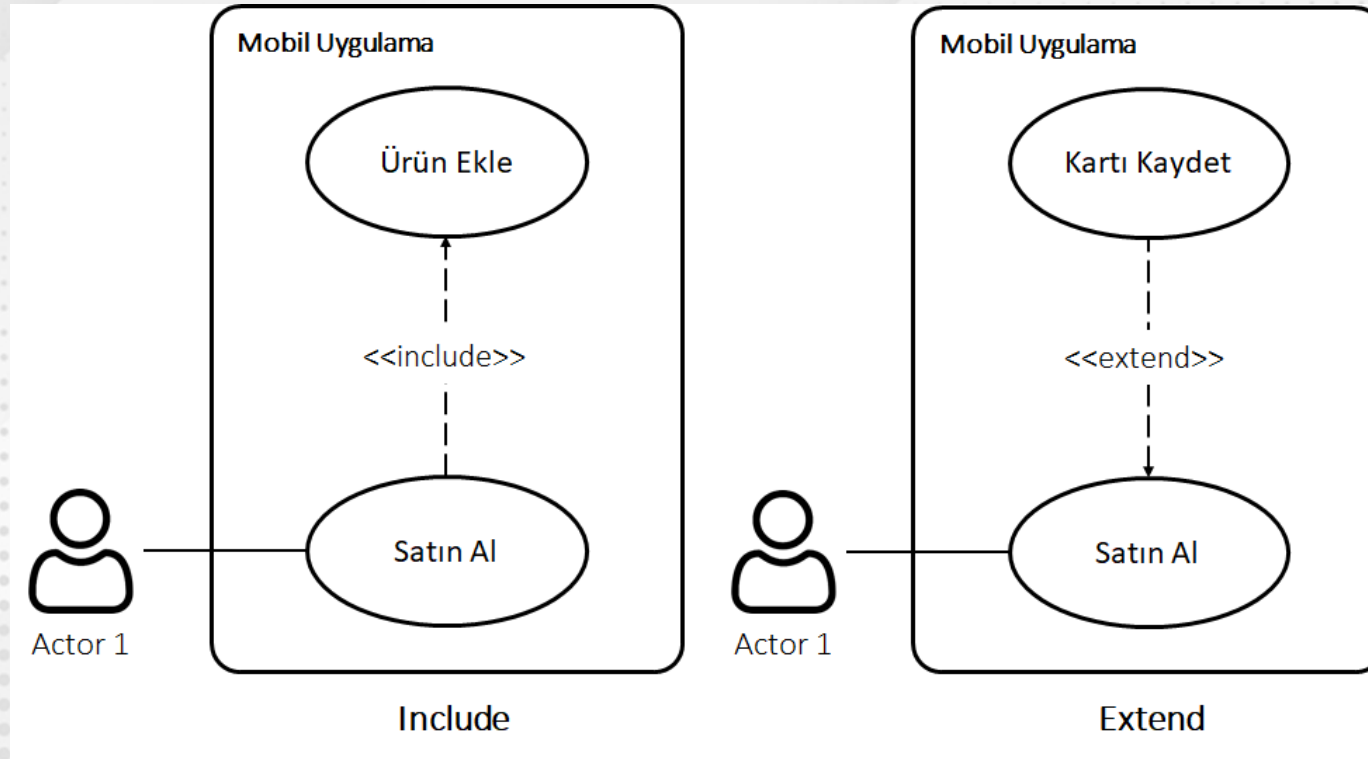
- * Aktörler belirlenir.
- * Her aktörün “ne” yapmak istediğı belirlenir.
- * Her aktörün “ne” si için “ana senaryo özeti” çıkarılır
- * Tüm sistemin ana senaryo özetleri incelenir, ayıklanır, birleştirilir.
- * Her senaryo için ana işlem adımları belirtilir.

Örnek Kullanım Senaryosu Diyagramı



<http://omerceitn.com.tr/DERS/AVP/Sunumlar/AvP03.pdf>

Örnek Kullanım Senaryosu Diyagramı

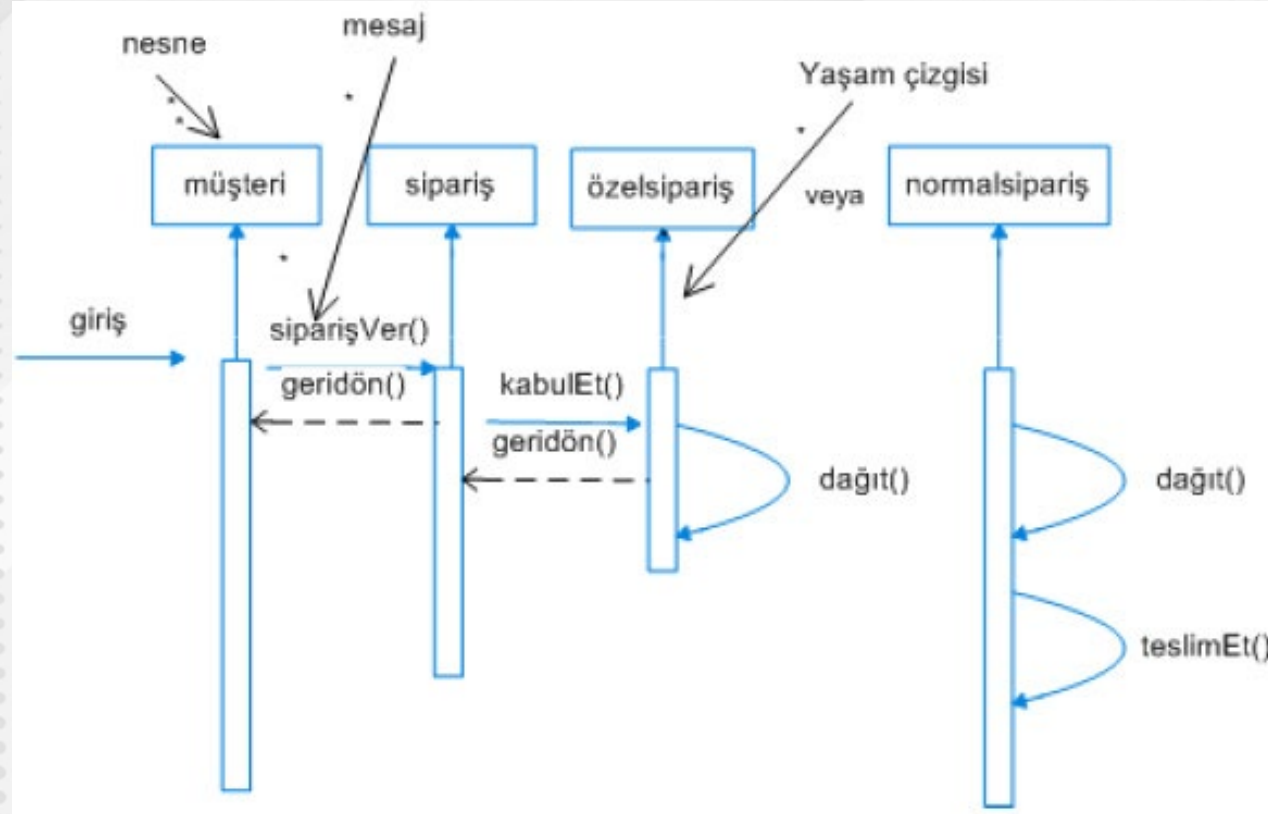


<http://omercetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>

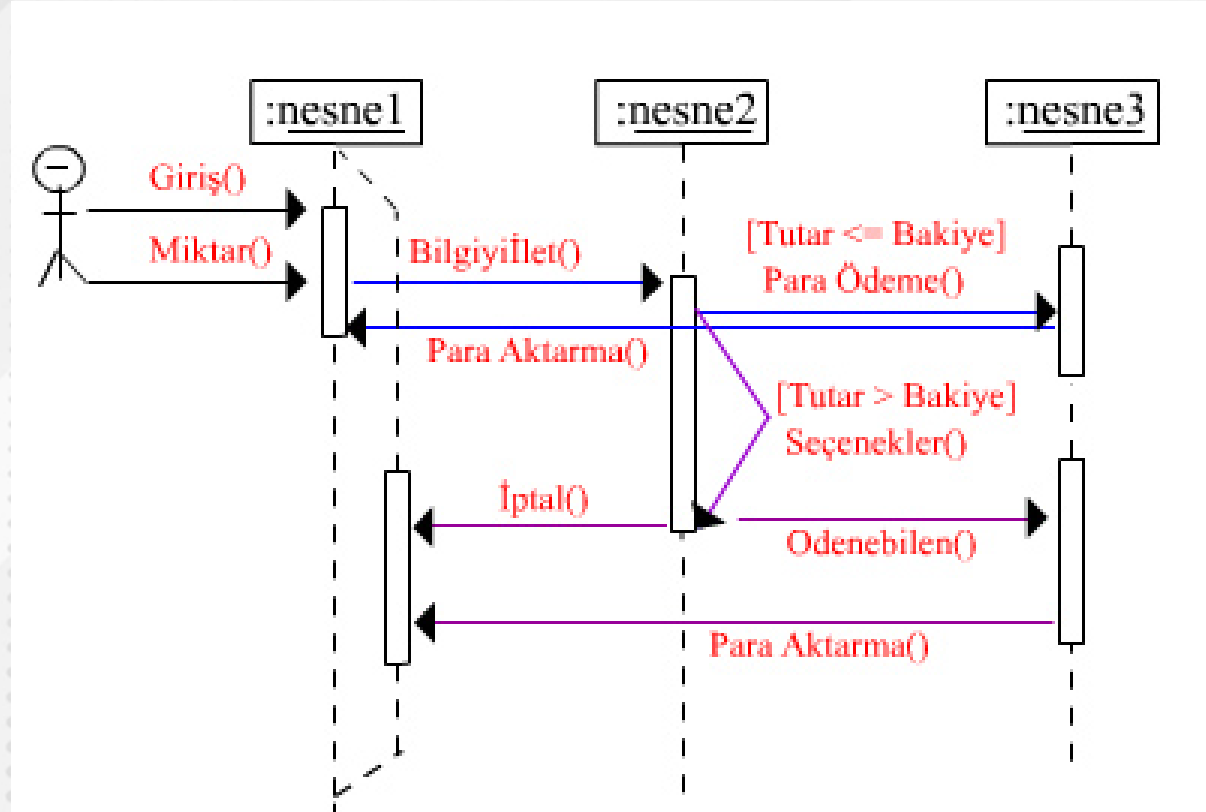
Ardışık Diyagramlar

- * Sistem içindeki nesnelerin zaman içindeki ardışık aksiyonlarını gösterirler.
- * Sistemin dinamik bir resmini çizer.
- * Aktörün hayat süresi boyunca gerçekleştirdiği işlemler gösterilir.
 - * Aksiyonlar->dikdörtgen
 - * Etkileşimler->mesajlar

Ardışık Diyagramlar

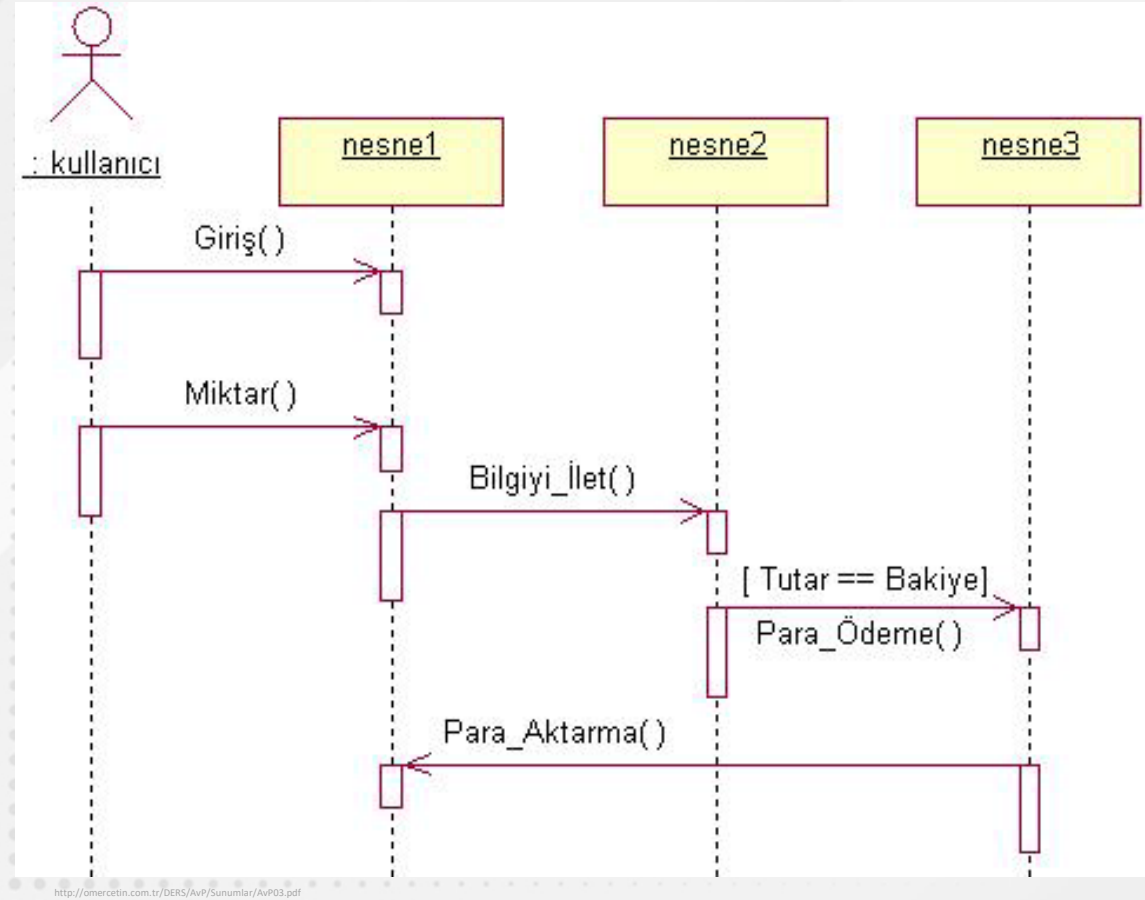
<http://omercetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>

Ardışık Diyagramlar



<http://omercetin.com.tr/DERS/AVP/Sunumlar/AvP03.pdf>

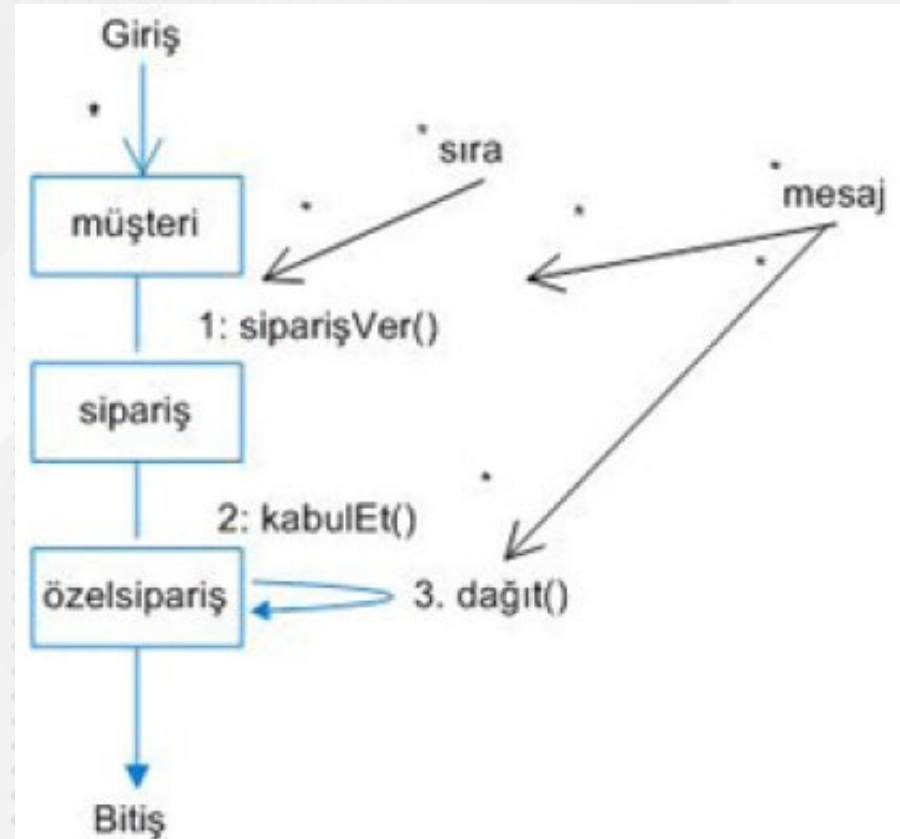
Ardışık Diyagramlar



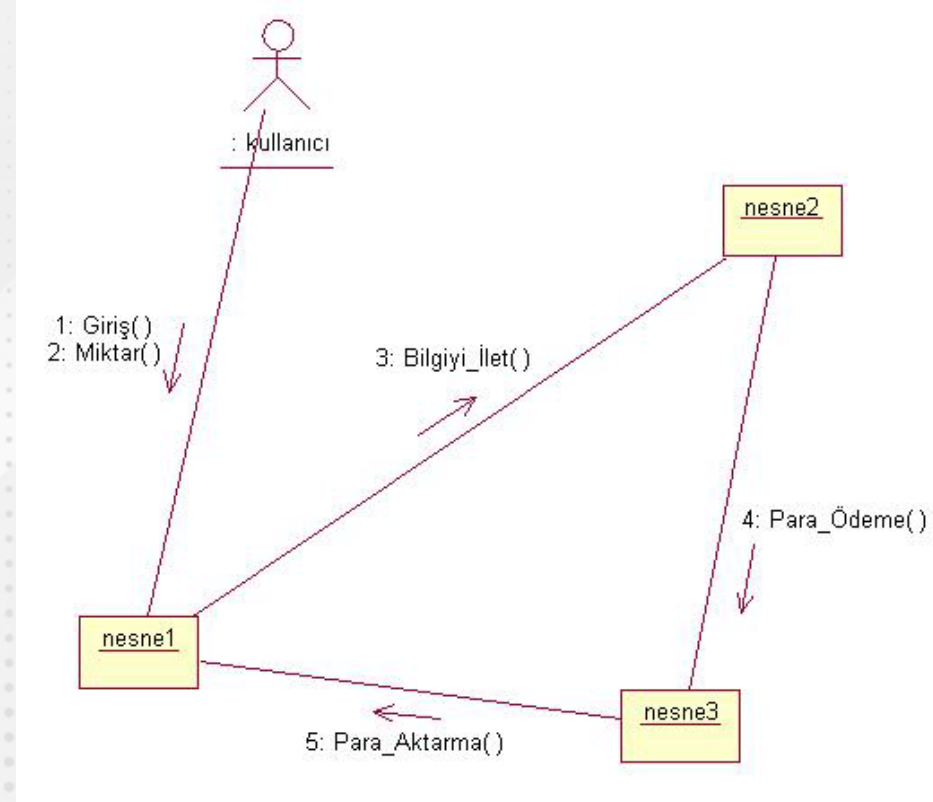
İşbirliği Diyagramları

- Parçaların bütünü nasıl oluşturduğunu ve sistemin dinamik davranışını göstermek amacıyla kullanılır.
- Nesneler arasındaki bağıntıları sıralı mesajlar şeklinde gösterirler.
- Ardışık diyagramlardan zaman kavramının olmayışı ile farklılık gösterirler.

İşbirliği Diyagramları

<http://omercetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>

İşbirliği Diyagramları



<http://www.csharpnedit.com/articles/read/?id=472&title=UML%20ile%20İşbirliği%20%28Collaboration%29%20Diyagramları>

İşbirliği Diyagramları ile Sıralama Diyagramları Arasındaki Farklar

1. Sıralama diyagramları, zamana bağımlı olarak nesneler arasındaki mesajlaşmayı göstermektedir. İşbirliği diyagramları ise zamandan bağımsızdır.
2. İşbirliği diyagramları ile tek bir diyagram üzerinde; dinamik yapıyı ve nesnelerin statik ilişkisini göstermeniz mümkündür. Sıralama diyagramları, sadece dinamik yapıyı temsil etmektedir.
3. Sıralama diyagramları, müşterilerin daha kolay anlayabileceği diyagramlardır. Bu nedenle; ilk analiz safhasında sıralama diyagramları kullanılırken, işbirliği diyagramları daha çok tasarım aşamasında kullanılır.

İşbirliği Diyagramları ile Sıralama Diyagramları Arasındaki Farklar

4. Sıralama diyagramlarında; mesajlaşma sıraları, işbirliği diyagramlarında ilişkiler daha kolay gözlemlenir. Büyük diyagramlar için; mesaj sırası ile mesajlaşmayı takip etmek (işbirliği diyagramları), şekil üzerinden takip etmeye (sıralama diyagramları) göre daha zordur. Ancak; daha az alanda birçok bilgiyi göstermenizi sağlayabilir.
5. Belirli bir amacı yerine getiren nesneler arasındaki ilişkiler, işbirliği diyagramları ile daha net gösterilebilmektedir. Senaryoya ilişkin tüm resmi (nesneler ve arasındaki ilişkiler) işbirliği diyagramları ile daha net gösterebilirsiniz. Sıralama diyagramları, mesaj sırasına yoğunlaştığı için tüm resim net görünmeyebilir.
6. Sıralama diyagramlarında bir bağlantı, bir nesneden diğerine giden mesajı gösterirken; işbirliği diyagramlarında bir bağlantı, nesneler arasındaki 1 veya daha fazla mesajı gösterebilir.

İşbirliği Diyagramları ile Sıralama Diyagramları Arasındaki Farklar

7. Sıralama diyagramlarının bir avantajı; nesnelerin yaratılışı ve sonlandırılışını kolay şekilde gösterebilmenize imkan vermesidir. Zaman boyutunda "yeni ()" mesajı ile nesneyi yaratabilir, X işareti ile de nesnenin sonlandırıldığını gösterebilirsiniz. İşbirliği diyagramlarında bu kadar basit olarak aynı işlemi gerçekleştiremezsiniz. Bu durumu gerçekleştirmek için "oluştur()" ve "sonlandır()" mesajlarını modele katmanız gerekir.

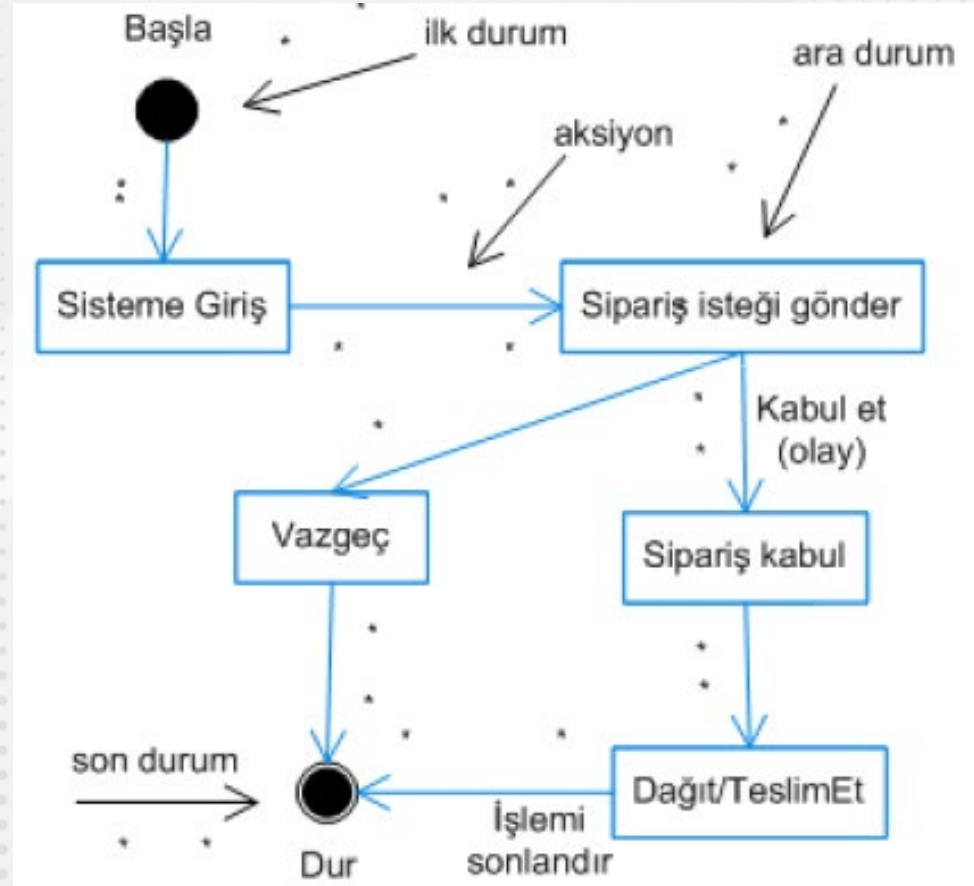
8. İşbirliği diyagramlarında zaman kavramı olmadığı için; bir nesnenin ne zaman aktif ne zaman pasif olduğunu mesaj tiplerini incelemeden anlayamazsınız ancak sıralama diyagramlarında bu durum çok kolay algılanmaktadır.

9. İşbirliği diyagramları; daha karmaşık dallanmalara (branch), eş zamanlı akış denetimine (concurrent flow of control) izin verirken sıralama diyagramları doğası gereği daha basit dallanmalara izin vermektedir. Bu açıdan; işbirliği diyagramları ile daha karmaşık sistemler modellenenbilmektedir.

Durum Diyagramları

- Sistemdeki nesnelerin anlık durumlarını göstermek amacıyla kullanılırlar.
- Sistemin küçük alt sistemlere veya nesnelere ilişkin dinamik davranışlarının ortaya çıkartılması amacıyla yararlanır.
- Sistemin durumları ve bunların birbirlerini tetikleme ilişkileri belirtilir.

Durum Diyagramları

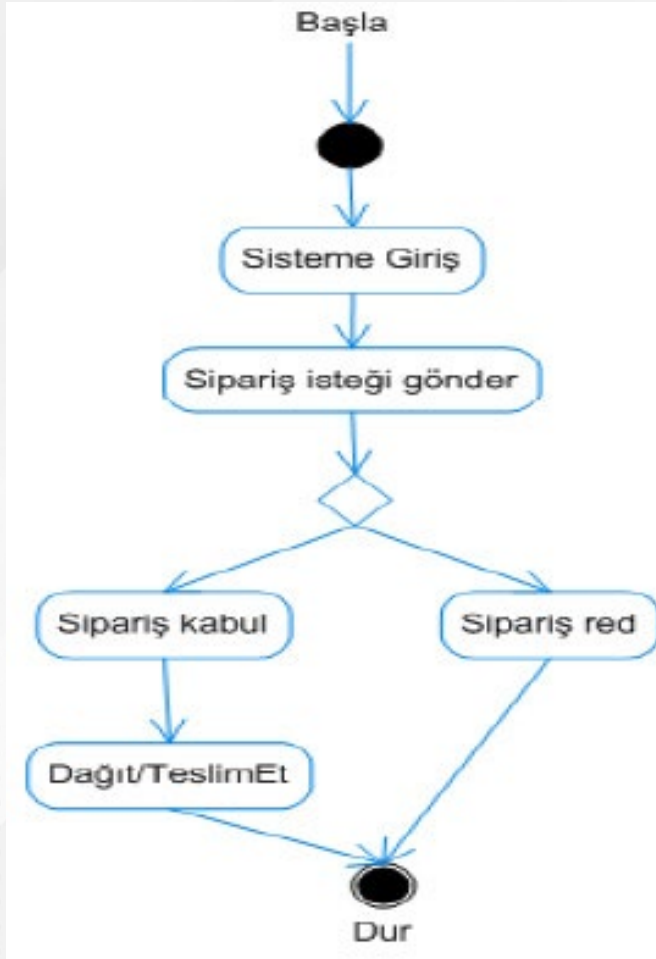


<http://omergetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>

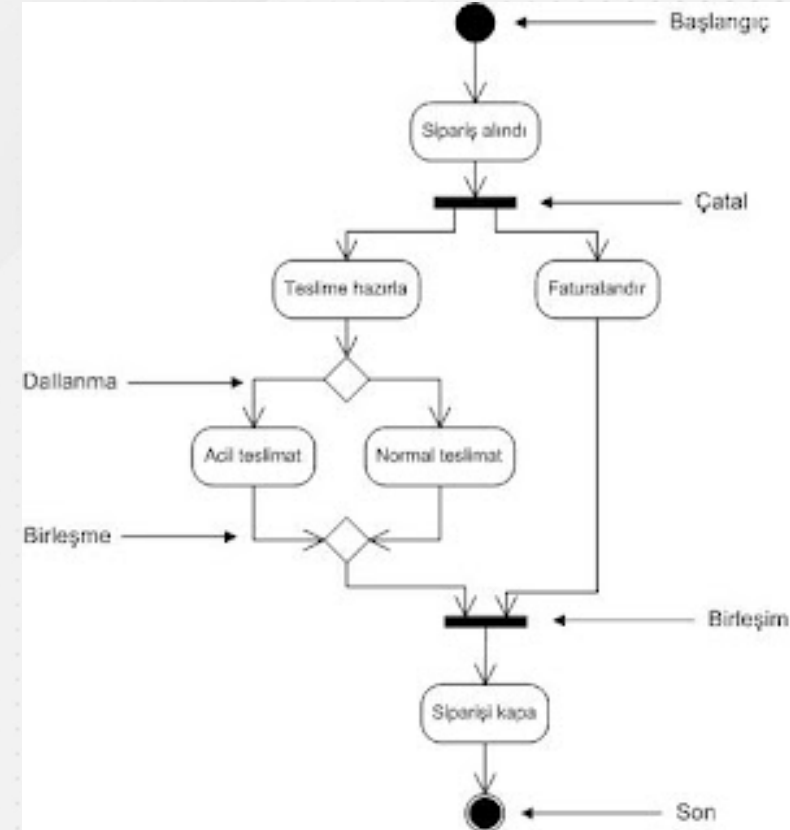
Etkinlik Diyagramları

- Sistemdeki nesnelerin faaliyetlerini göstermek amacıyla kullanılırlar.
- Durum diyagramlarının bir alt kümesi olarak değeriendirilen etkinlik diyagramları, iş akışlarının grafiksel gösteriminde kullanılırlar.

Etkinlik Diyagramları



<http://omercetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>



<http://omercetin.com.tr/DERS/AvP/Sunumlar/AvP03.pdf>

Kaynakça

- Algan, S. (2002). <http://www.csharpnedir.com/articles/read/?id=6>
- Kurt, B. Ders notları:
http://web.itu.edu.tr/bkurt/Courses/oose/ooswe_release_1_0_6.pdf
- Bozkır, S. <http://www.slideshare.net/aselmanb/uml-ile-modelleme>
- Tokdemir, G. ve Çağıltay, N. E. (2010). *Veritabanı Sistemleri Dersi*. Seçkin yayıncılık, Ankara.