

Nesneye Yönelik Tasarım

BBS-651 Yazılım Mühendisliği
Hafta #5 (3 Kasım 2010)

İçerik

- Nesneye yönelik tasarım
 - ▶ Nesne ve sınıf kavramları
 - ▶ Tasarım için kullanılan başlıca UML elemanları
 - ▶ Üst düzey tasarım adımları
 - ▶ Detay tasarım adımları
- Örnek çözümleme
 - ▶ Kütüphane destek sistemi
- Ödev 2
 - ▶ KDS için örnek UML diyagramlarının hazırlanması

A.Tarhan, 2010

- 2 -

BBS-651-DN05

“Nesne” ve “Sınıf” Kavramları

“Nesne” Nedir?

- İyi tanımlı bir kapsamı ve kimliği olan, belirli bir durum ve davranışı içeren, soyut veya somut varlıktır.
 - ▶ “A discrete entity with a well-defined boundary and identity that encapsulates state and behavior”
 - ▶ Nesne gerçek dünyadaki somut bir varlığı temsil edebilir.
 - Televizyon, motor, vb.
 - ▶ Nesne tamamen kavramsal bir varlığı temsil edebilir.
 - Banka hesabı, vb.

A.Tarhan, 2010

- 4 -

BBS-651-DN05

“Nesne”: Örnek

Müşteri Hesabı

miktar

paraÇek
ParaYatır
miktarSorgula

- Her nesne aşağıdakilere sahiptir:
 - ▶ Kişilik (“identity”)
 - ▶ Özellik (“attribute”)
 - ▶ Durum (“state”)
 - ▶ Davranış (“behavior”)
 - ▶ Sorumluluk (“responsibility”)

A.Tarhan, 2010

- 5 -

BBS-651-DN05

Özellik

- Her nesnenin kendine ait bir dizi özelliği vardır.
 - ▶ Özellikler nesneye ait verileri taşır.

:Arabam

model
marka
renk

:BirMusteri

isim
musteriNo
bakiye

:BirPencere

boyut
pozisyon
renk

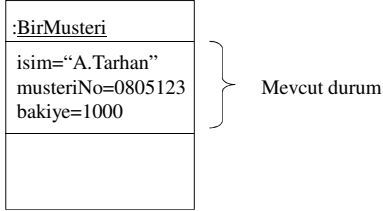
A.Tarhan, 2010

- 6 -

BBS-651-DN05

Durum (1)

- Nesnenin tüm özellikleri ve bu özelliklerin o anki değerleri, nesnenin durumunu oluşturur.



A.Tarhan, 2010

- 7 -

BBS-651-DN05

Davranış

- Her nesnenin iş yapabilmeyi sağlayan tanımlı bir davranış şekli vardır.

- ▶ Nasıl davranır, nasıl tepki verir ? ("act"/"react")
 - Nesnenin operasyonları
 - Görülebilir aktivite
 - Diğer nesneler tarafından kullanılan arayüz ("public interface-operations")
- ▶ Davranışın gerçekleştirilmesi bilgisayar kodu ile yapılır.
 - Kodlanan nesne yordamları, davranışı gerçekleştirir.
 - Gizli gerçekleştirme ("encapsulated implementation")

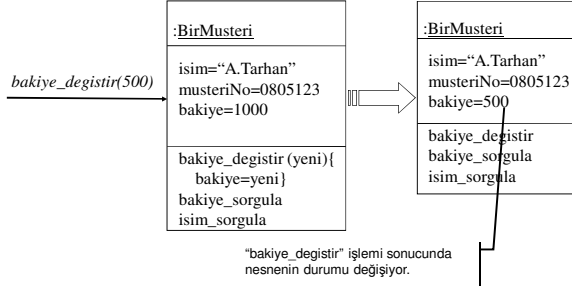
A.Tarhan, 2010

- 8 -

BBS-651-DN05

Durum (2)

- İşlemler sonucunda nesnenin durumu değişir.



A.Tarhan, 2010

- 9 -

BBS-651-DN05

Sorumluluk

- Nesnenin sorumluluğu tüm sistemin işlevselliğine nasıl katkıda bulunacağını tanımlar.

- ▶ Nesnenin sistemde oynayacağı rol
- ▶ Neden böyle bir nesneye gereksinim var ?
- ▶ Özellikler ve davranışlar, birlikte nesnenin sorumluluklarını yerine getirmesini sağlar.

A.Tarhan, 2010

- 10 -

BBS-651-DN05

İlişki ("Relationship")

- Nesneler arasındaki fiziksel veya kavramsal bağlantı
- En yaygın ilişki: Bir nesne diğer nesnenin sunduğu servislerden yararlanır.
 - ▶ Mesaj iletimi
 - ▶ Müşteri/tedarikçi ilişkisi
- Sistem işlevselliğini sağlamak amacıyla, nesneler birbirleriyle ilişkiler aracılığı ile işbirliği yaparlar.
 - ▶ Nesneye yönelik programlama modeli

A.Tarhan, 2010

- 11 -

BBS-651-DN05

"Sınıf" Nedir?

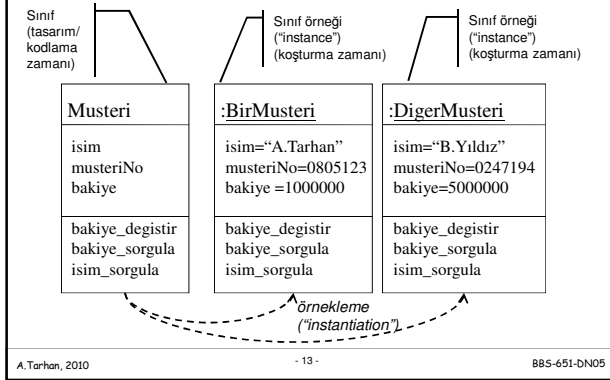
- Yapısal ve/veya davranışsal olarak aynı özelliklere sahip nesneler SINIF altında gruplanır.
 - ▶ Her nesne bir sınıfın örneğidir ("instance").
 - ▶ Sınıfları tanımlar ve nesneleri sınıf tanımından örnekleriz ("instantiation").
 - ▶ Her nesne ait olduğu sınıfı bilir.
- Sınıf, nesneler için şablon tanımdır.
 - ▶ Özellikler ve yordamlar sınıf için yalnızca bir kez tanımlanır.
- Sınıfların birbirleri arasındaki ilişkiler sistemin sınıf yapısını ("class structure") oluşturur.
 - ▶ İki sınıf arasında ilişki varsa karşılık gelen nesneler arasında da vardır.

A.Tarhan, 2010

- 12 -

BBS-651-DN05

“Sınıf”: Örnek



“Sınıf” ve “Nesne”

- Her sınıfın sıfır veya daha fazla örneği vardır.
- Sınıf statik, nesne dinamikdir.
 - Sınıfın varlığı, semantiği ve ilişkileri program koşuturulmadan önce sabit olarak belirlenmiştir.
 - Nesneler program koşuturulduğunda sınıf tanımından dinamik olarak yaratılırlar ("construction").
 - Nesneler sorumluluklarını tamamladıklarında ortadan kaldırılırlar ("destruction").
- Nesnenin sınıfı sabittir ve nesne bir kez yaratıldıktan sonra değiştirilemez.

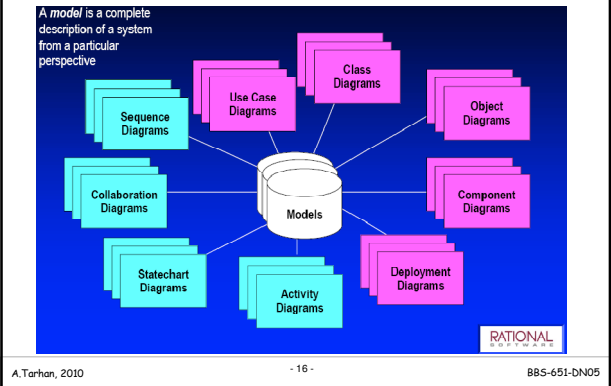
A.Tarhan, 2010

- 14 -

BBS-651-DN05

Nesneye Yönelik Tasarım İçin Kullanılan Başlıca UML Elemanları

UML Diyagramları



Tasarım İçin Kullanılan Başlıca UML Elemanları (1)

- Sınıf diyagramı ("class diagram")
 - Sistemi oluşturan sınıflar ve bunlar arasındaki ilişkiler
- Ardıl-işlem diyagramı ("sequence diagram")
 - Nesneler arası etkileşim (davranış)
 - Nesneler arası kontrolün akışı (zamana göre sıralı)
- İşbirliği diyagramı ("collaboration diagram")
 - Nesneler arası etkileşim (davranış)
 - Nesneler arası kontrolün akışı (mesaj sıralı)

A.Tarhan, 2010

- 17 -

BBS-651-DN05

Tasarım İçin Kullanılan Başlıca UML Elemanları (2)

- Durum diyagramı ("statechart diagram")
 - Dinamik davranış, olay ("event") esaslı
 - Bir nesnenin iç davranışı (nesne yaşam döngüsü)
- Etkinlik diyagramı ("activity diagram"):
 - Operasyon akışları
- Bileşen diyagramı ("component diagram"):
 - Uygulamanın fiziksel yapısı
 - Gerçekleştirme aşamasında
- Yayılma diyagramı ("deployment diagram"):
 - Donanım topolojisi
 - Tasarım ve gerçekleştirme aşamalarında

A.Tarhan, 2010

- 18 -

BBS-651-DN05

Üst Düzey Tasarım

Üst Düzey Tasarım

- Amaç: Sistemi oluşturacak temel varlıkları ve ilişkilerini tanımlamak
 - ▶ Sistemin sınıf yapısı
 - Uygulama alanına özgü sınıflar ("domain classes")
 - Sınıflar arası ilişkiler
 - ▶ Davranış modeli
 - Sınıfların işlevsel gereksinimleri karşılamak için birbirleri ile nasıl işbirliği yapacakları
 - ▶ Literatürde "OO Analysis" olarak da geçiyor

A.Tarhan, 2010

- 20 -

BBS-651-DN05

"Use Case" ve Tasarım Modelleri

USE-CASE MODELİ :

- Doğal dil
- Dış bakış açısı ("black box")
- "Use case" ile yapılandırılır
- Müşteriler esaslı
- Tekrar ve tutarsızlıklar olabilir
- İşlevselliğin "ne" olduğunu tanımlar

TASARIM MODELİ :

- Yazılım modelleme dili
- İç bakış açısı
- Sınıflar ile yapılandırılır
- Geliştiriciler esaslı
- Tekrarlar ve tutarsızlıklar olamaz
- İşlevselliğin sistemde "nasıl" gerçekleştirileceğini tanımlar

A.Tarhan, 2010

- 21 -

BBS-651-DN05

Üst Düzey Tasarım Adımları

1. Sınıfları belirle
 - ▶ Amaç: "Use case" tanımlarından analiz sınıflarını ("domain classes") ve sorumluluklarını belirlemek
 - Odak noktası sınıfları belirlemektir (detaylı tanımların sonra yapılması önerilir)
2. Sınıf yapısını belirle
 - ▶ Amaç: Belirlenen sınıfların "use case"lerde tanımlanan iş adımlarını gerçekleştirmek için nasıl etkileşimde bulunacağını saptamak
 - "Use case"ler gözden geçirilerek belirlenen sınıfların arasındaki ilişkiler tanımlanır
3. Davranış modelle
 - ▶ Amaç: "Use case"lerin içerdiği işlevselliğin, belirlenen sınıfların davranışları aracılığıyla nasıl gerçekleştirileceğini tanımlamak
 - Nesnelerin belirlenen işi yerine getirmek için nasıl mesajlaşacakları belirlenir
 - Nesneler arası kontrol akışı dinamik olarak gösterilir

A.Tarhan, 2010

- 22 -

BBS-651-DN05

Adım-1. Sınıfları Belirle (1)

- Use case modelinde yer alan her use case tanımı gözden geçirilir
 - ▶ Analiz sınıfı ("domain class") özelliklerini taşıyan varlıklar belirlenir
 - ▶ Belirlenen sınıflara sorumluluk atanır
 - Use case tanımlarındaki işler esas alınır
 - ▶ Önceki use case'lerde belirlenmiş sınıfları kullanmaya çalışınız
 - Analiz sınıfları büyük olasılıkla birçok use case'de karşımıza çıkacaktır

A.Tarhan, 2010

- 23 -

BBS-651-DN05

Adım-1. Sınıfları Belirle (2)

- Use case tanımlarının üzerinden geçerken:
 - ▶ Kullanılan "isimler" aday sınıf olarak ele alınır
 - ▶ Aday sınıflar belirli kurallar ve gereksinimler doğrultusunda elenir
 - ▶ Elemler sonucunda oluşan "isim" kümesi sistemin sınıfları olarak ele alınır
 - ▶ Elemlerden geçemeyen "isim"ler belirlenen sınıfların özellikleri olabilirler
 - ▶ İş tanımlayan "fiil"ler sınıfların sorumluluklarını veya sınıflar arası ilişkileri tanımlar

A.Tarhan, 2010

- 24 -

BBS-651-DN05

Adım-1. Sınıfları Belirle (3)

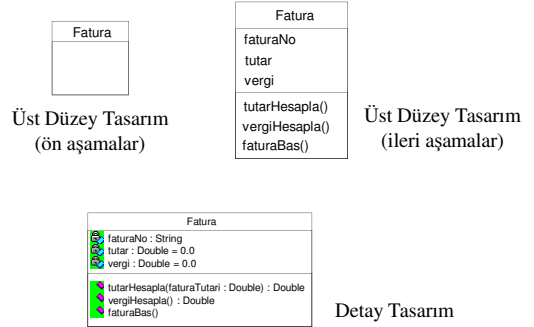
- Belirlenen her sınıf için, sınıfın özellikleri tanımlanır:
 - Özellikler "use case" tanımlarında doğrudan yer alabilir
 - Özellikler sınıfın sorumluluklarından çıkarılabilir
 - Bu aşamada özellikler ile ilgili detaylı bir tanımlamaya gerek yoktur
 - Veri yapılarına girilmeden sadece isimlendirilir
- Belirlenen her sınıf için sınıfın davranışı, sorumluluklar esas alınarak operasyon halinde tanımlanır
 - "Use case"de tanımlanan işlevsellik, sınıflar arasında paylaşılır
 - Sorumluluk, sınıfın farklı "use case"lerde aldığı rollerin birleşimidir
 - Her rol bir operasyon olabilir
 - Bu aşamada "class methods", "signatures" detaylarına girilmez

A.Tarhan, 2010

- 25 -

BBS-651-DN05

Tasarım Boyunca Sınıf Tanımının Gelişimi: Örnek



A.Tarhan, 2010

- 26 -

BBS-651-DN05

Adım-2. Sınıf Yapısını Belirle

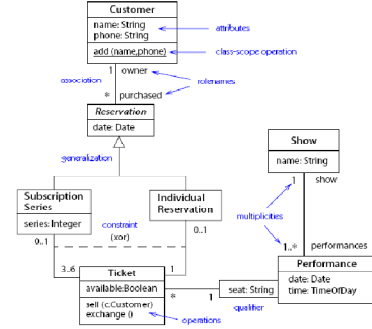
- Belirlenen ilişkiler davranış modellemesine esas oluşturur
 - "Fiil"ler ilişkilerin göstergesidir
- Sınıflar arası ilişkiler kavramsal olarak tanımlanır
 - İlişkinin türlerine ve detay özelliklerine girilmez
- Sınıflar arası ilişkiler en az sayıda tutulmaya çalışılır
- Tasarım süreci devam ettikçe basit ilişkiler özel formlara dönüşebilir
 - "association", "aggregation", "composition", "generalization", vb.

A.Tarhan, 2010

- 27 -

BBS-651-DN05

"Static View" "Class Diagram": Örnek



A.Tarhan, 2010

- 28 -

BBS-651-DN05

Sınıf Diyagramı

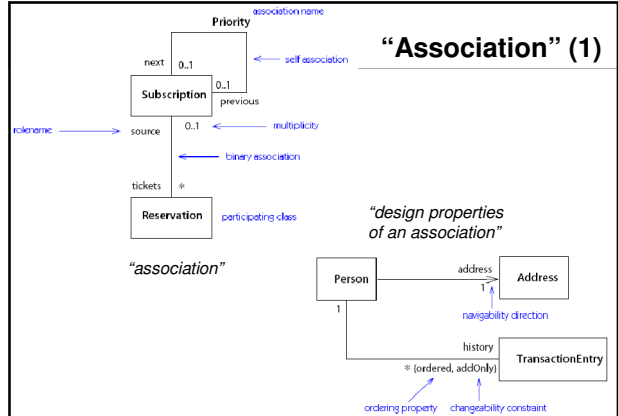
- Sınıf diyagramı "statik" bakış açısı sunar
 - Yol haritası gibi
 - Nesneler şehirleri, ilişkiler şehirler arasındaki yolları gösterir
 - Hedefe ulaşmak için hangi yolun takip edilmesi gerektiğini söylemez
- Sınıf diyagramında hangi nesnelerin işbirliği yaptığı belli, ancak nesnelerin nasıl işbirliği yapacakları belli değildir
 - Nasıl sorusunun cevabı "sequence" veya "collaboration" diyagramlarında tanımlanır
- Sınıf diyagramları geliştirme boyunca kullanılır
- Sınıf diyagramı, sistem için tanımlanan tüm sınıfları içermeyebilir
 - Sistem modeli farklı sınıf diyagramlarından oluşabilir
 - Her sınıf diyagramının bir görevi vardır
 - "Collaboration", "generalization", vb.

A.Tarhan, 2010

- 29 -

BBS-651-DN05

"Association" (1)

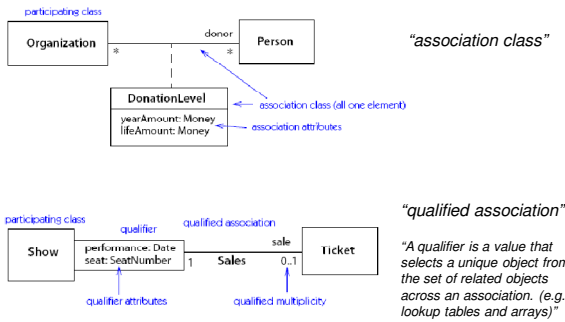


A.Tarhan, 2010

- 30 -

BBS-651-DN05

“Association” (2)



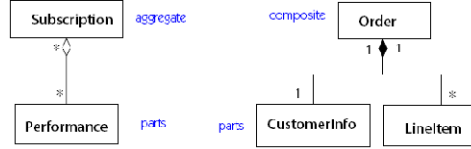
A.Tarhan, 2010

- 31 -

BBS-651-DN05

“Aggregation” ve “Composition”

■ Bütün-parça ilişkisi



"An aggregation is an association that represents a part-whole relationship."

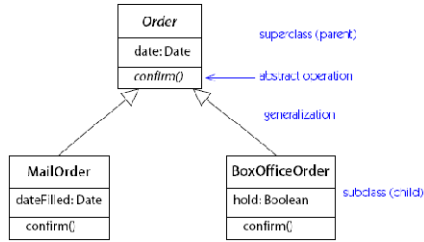
"A composition is a stronger form of association in which the composite has sole responsibility for managing its parts—such as their allocation and deallocation."

A.Tarhan, 2010

- 32 -

BBS-651-DN05

“Generalization”



A.Tarhan, 2010

- 33 -

BBS-651-DN05

Adım-3. Davranış Modelle

■ Davranışları iki şekilde tanımlanabilir:

- ▶ Ardıl-işlem ("sequence") diyagramları ile
 - Nesneler arası iletişimin zamana göre sıralaması
- ▶ İşbirliği ("collaboration") diyagramları ile
 - Nesnelerin etkileşimde aldığı roller ve ilişkiler
 - Nesneler arası iletişimin etkileşime göre sıralaması

■ Önerilen yöntem:

- ▶ Bir "use case" veya senaryo seçilir
- ▶ Hangi nesnelerin rol alacağı belirlenir
- ▶ Bir nesneden diğerine gönderilecek mesajlar zamana veya etkileşime göre sıralanarak gösterilir

A.Tarhan, 2010

- 34 -

BBS-651-DN05

Ardıl-İşlem ("Sequence") Diyagramı

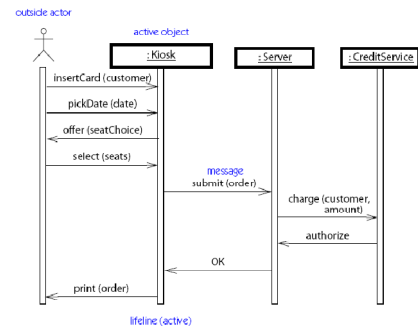
- Nesneler arasındaki ilişkileri zaman sırasına göre düzenlenmiş olarak göstermeye yarar
- Bir etkileşimde yer alan nesneleri ve çağrılan mesajların sırasını gösterir
- İşbirliği diyagramı ile benzer bilgileri farklı biçimde gösterir
- İşbirliği diyagramında bulunan nesne ilişkileri ardıl-işlem diyagramında açıkça görülmez

A.Tarhan, 2010

- 35 -

BBS-651-DN05

“Interaction View” “Sequence Diagram”: Örnek

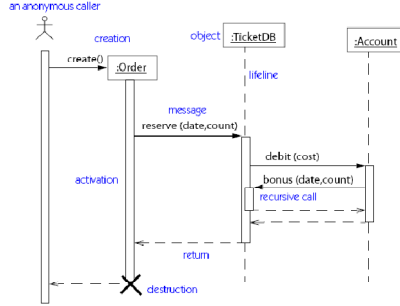


A.Tarhan, 2010

- 36 -

BBS-651-DN05

“Activation”



A.Tarhan, 2010

- 37 -

BBS-651-DN05

İşbirliği (“Collaboration”) Diyagramı

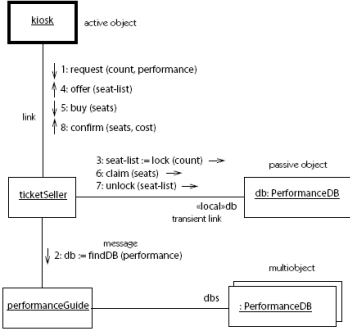
- İşbirliği ve ardıl-işlem diyagramları nesneleri ve aralarındaki mesajlaşmayı dinamik olarak modellemeye yarar
- İşbirliği diyagramı, ardıl-işlem diyagramından farklı olarak, sınıflar arasındaki ilişkileri doğrulamada yardımcı olur
 - Fazla ya da eksik ilişkileri bulabilirsiniz

A.Tarhan, 2010

- 38 -

BBS-651-DN05

“Interaction View” “Collaboration Diagram”: Örnek



A.Tarhan, 2010

- 39 -

BBS-651-DN05

Örnek Çözümleme: Kütüphane Destek Sistemi

- Kütüphane işlemlerini desteklemek amacıyla bir yazılım sistemi oluşturulacaktır.
- Sistem; kayıtlı müşterilere, yine kayıtlı kitap ve dergileri ödünç verecektir.
- Kütüphane, yeni başlıklı kitap ve dergilerin satın alınmasını yapacaktır. Popüler başlıklar, birden çok kopya satın alınacaktır. Eski kitap ve dergiler, zaman aşımına uğradıklarında veya çok yıprandıklarında yok edilecektir.
- Kütüphanede, müşterilerle iletişimi sağlayacak ve yaptığı işler sistem tarafından desteklenecek bir kütüphane görevlisi bulunacaktır.
- Müşteri, kütüphanede o anda bulunmayan bir kitap veya dergiyi rezerve edebilecektir. Kitap veya dergi kütüphaneye geri döndürüldüğünde, rezervasyonu yapan müşteri haberdar edilecektir. Rezervasyon, müşteri kitap veya dergiyi ödünç aldığı anda veya müşterinin özel isteği üzerine iptal edilecektir.
- Sistem; kitap ve dergi başlıklarının, kitap ve dergi kopyalarının, müşterilerin, ödünç işlemlerinin ve rezervasyonların kaydedilmesine, güncellenmesine ve silinmesine olanak sağlayacaktır.
- Sistem tüm popüler bilgisayar ortamlarında (UNIX, Windows, OS/2, vb.) çalışacak ve modern bir kullanıcı ara yüzüne sahip olacaktır.
- Sistem yeni işlevler eklemek suretiyle genişletilebilir olacaktır.

A.Tarhan, 2010

- 40 -

BBS-651-DN05

Örnek Çözümleme: KDS

Detaylı Tasarım

A.Tarhan, 2010

- 41 -

BBS-651-DN05

Detaylı Tasarım

- Amaç: Tasarım modellerini, gerçekleştirmeye olanak sağlayacak şekilde detaylandırmak
 - ▶ Odak gerçekleştirme detaylarına kayar
 - ▶ Üst düzeyde oluşturulan UML diyagramları somut kararlar alındıkça zenginleştirilir
 - ▶ Sistem mimarisini ve detay davranışı modellemek için farklı UML diyagramları oluşturulur

A.Tarhan, 2010

- 43 -

BBS-651-DN05

Ele Alınacak Hususlar

- Gereksinimlerde tanımlanan işi gerçekleştirmek için gereken diğer sınıflar
- Gerçekleştirmede kullanılacak algoritmalar ve veri yapıları
- Kullanılacak tasarım örüntüleri
- Kullanılacak hazır parçalar ("component")
- Veri saklama yöntemleri (ilişkisel veri tabanı, vb.)
- Uygulamanın üzerinde çalışacağı platform
 - ▶ Ortam dağıtık olacak mı ?
 - ▶ Uygulama nasıl parçalanacak ?
- Performans kısıtları
- ...

A.Tarhan, 2010

- 44 -

BBS-651-DN05

Detaylı Tasarım Adımları

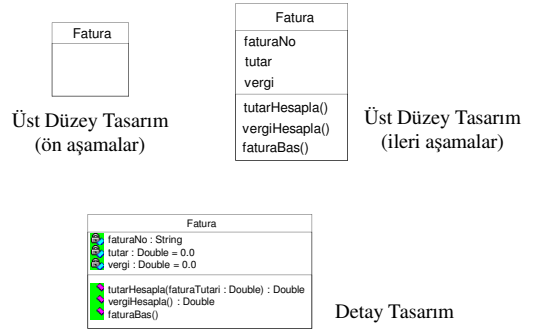
1. Sınıf yapısını detaylandır
 - ▶ Amaç: Sınıf diyagramlarını, gerçekleştirmeye yönelik zenginleştirmek
 - Sınıf tanımlarını ve ilişkileri detaylandırmak ve/veya kısıtları belirlemek
 - Tasarım sınıflarını ve aralarındaki ilişkileri belirlemek
 - Tasarım sınıflarının diğer sınıflarla ilişkilerini belirlemek
2. Davranış modelini detaylandır
 - ▶ Amaç: Sınıfların ve sistemin davranışını daha detaylı olarak modellemek
 - Nesneler arası işbirliği kapsamında
 - Ardıl-işlem ("sequence") diyagramı; İşbirliği ("collaboration") diyagramı
 - Bir nesne kapsamında
 - Durum ("statechart") diyagramı; Etkinlik ("activity") diyagramı
3. Mimariyi modelle
 - ▶ Amaç: Yazılım sistemini parçaları ("components") cinsinden tanımlamak
 - Diyagramları ve model varlıklarını, paket ("package") yapısı altında gruplandırmak

A.Tarhan, 2010

- 45 -

BBS-651-DN05

(Adım-1. Sınıf Yapısını Detaylandır) Tasarım Boyunca Sınıf Tanımının Gelişimi: Örnek

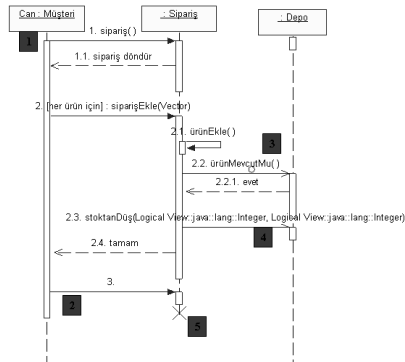


A.Tarhan, 2010

- 46 -

BBS-651-DN05

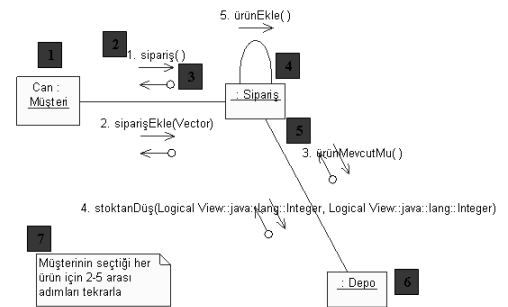
(Adım-2. Davranış modelini detaylandır) Ardıl-işlem Diyagramı: Örnek



A.Tarhan, 2010

BBS-651-DN05

(Adım-2. Davranış modelini detaylandır) İşbirliği Diyagramı: Örnek



A.Tarhan, 2010

- 48 -

BBS-651-DN05

**(Adım-2. Davranış modelini detaylandır)
Durum ("Statechart") Diyagramı**

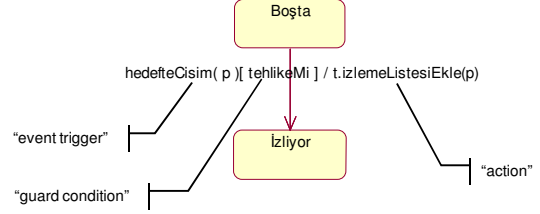
- Nesnelerin dinamik davranışlarını tanımlar
- Davranışı karmaşık olan nesneleri modellemek için kullanılır
- Bir nesnenin farklı durumları ("state") arasındaki geçişleri modeller
- Özellikle **reaktif nesnelerin** davranışlarının modellenmesinde etkilidir
 - Asenkron olaylara göre davranış gösteren nesneler
- Sistemin genel davranışının modellenmesi için de kullanılabilir

A.Tarhan, 2010

- 49 -

BBS-651-DN05

**(Adım-2. Davranış modelini detaylandır)
Durum Diyagramı: Örnek**



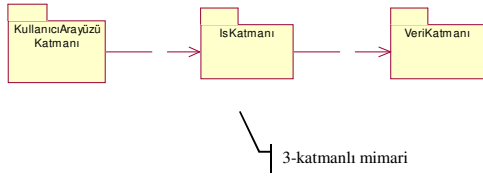
A.Tarhan, 2010

- 50 -

BBS-651-DN05

**(Adım-3. Mimariyi Modelle)
Kavramsal Mimari**

- Uygulamanın fiziksel platformdan bağımsız, farklı katmanlar halinde tasarlanması
 - Tek bir platformda çalışacak bile olsa farklı katmanlardan oluşmasında fayda var



3-katmanlı mimari

A.Tarhan, 2010

- 51 -

BBS-651-DN05

**(Adım-3. Mimariyi Modelle)
Fiziksel Mimari**

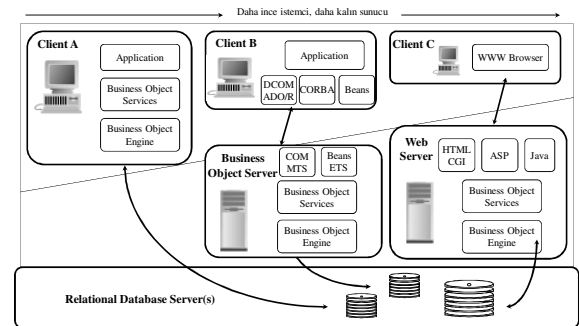
- Uygulamanın fiziksel olarak nasıl parçalanacağını tanımlar
 - Kavramsal mimari ile aynı olabilir
 - Kavramsal mimariden farklı olabilir
- Örnekler:
 - 3-katmanlı mimari
 - Kullanıcı arayüzü: Client
 - İş katmanı: Uygulama sunucusu
 - Veri katmanı: Veri sunucusu
 - 2-katmanlı mimari
 - Kullanıcı arayüzü: Client
 - İş ve Veri katmanları: Uygulama sunucusu

A.Tarhan, 2010

- 52 -

BBS-651-DN05

**(Adım-3. Mimariyi Modelle)
Fiziksel Mimari: Örnek**



A.Tarhan, 2010

- 53 -

BBS-651-DN05

Bileşen ("Component") Diyagramı

- Temel kullanımı:
 - Fiziksel yazılım bileşenlerini ve aralarındaki ilişkileri modellemek
 - Kaynak kodlar ve dosyalar arasındaki ilişkileri modellemek
 - Yazılım sürümlerinin yapısını modellemek
 - Çalıştırılabilir kod oluşturan kaynak dosyalar arasındaki ilişkileri modellemek
- UML'de bileşen ("component"):
 - Bileşen, işleyen sistemi oluşturan parçalardan biridir
 - İşleyen kod parçası, kütüphane, tablo, dosya, vb.
 - Sistemin fiziksel ve değiştirilebilen bir parçasıdır
 - Aynı arayüze sahip farklı parçalar ile değiştirilebilir
 - Bir veya birden fazla sınıf veya paket içerebilir



A.Tarhan, 2010

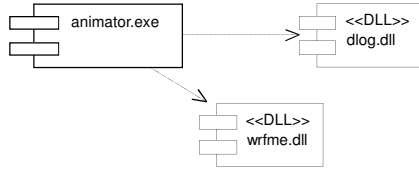
- 54 -

BBS-651-DN05

Bileşen Diyagramı: Örnek

■ Gösterim:

- Bileşenler ve aralarındaki ilişkileri gösteren oklar
- Okun yönü bağımlı bileşenden, bağımlı olduğu bileşene doğru
- İlişki okları "stereotype" kullanarak isimlendirilebilir
 - Örnek: "include", "derive", "friend", "import", vb.



A.Tarhan, 2010

- 55 -

BBS-651-DN05

Yayılma ("Deployment") Diyagramı

■ Bir yazılım sisteminde yer alan donanım öğelerini ve aralarındaki ilişkileri modeller

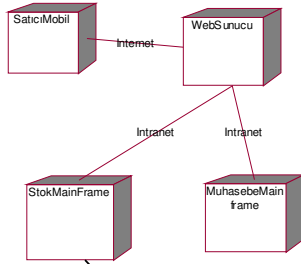
- Yayılma diyagramında iletişim ilişkisi ile birleştirilmiş "node"lar yer alır
 - "Node"lar sistemdeki işlemci kaynakları temsil eder ve küp şeklinde gösterilir
 - Bilgisayarlar, alıcılar, çevresel veya gömülü sistemler, vb.
 - İletişim ilişkileri "stereotype" kullanılarak iletişimin tipini gösterebilir

A.Tarhan, 2010

- 56 -

BBS-651-DN05

Yayılma Diyagramı: Örnek



A.Tarhan, 2010

- 57 -

BBS-651-DN05

Ödev-2

Kütüphane Destek Sistemi için;

1. Sistemin aktörlerini ve "use case"lerini belirleyerek "use case diagram" çizin.
2. Seçtiğiniz bir "use case"nin akışını "activity diagram" kullanarak gösterin.
3. Sistemin analiz sınıflarını ve aralarındaki ilişkileri belirleyerek "class diagram" çizin.
4. İkinci adımda detaylandırdığınız "use case"nin gerçekleştirilmesi için tasarladığınız "sequence diagram"ı çizin.

■ Önerilen Araçlar:

- Visual Paradigm for UML
- Rational Application Developer

■ Teslim: 23 Kasım 2010 24:00, atarhan@cs.hacettepe.edu.tr adresine e-posta ile

A.Tarhan, 2010

- 58 -

BBS-651-DN05

2. Arasınan Ödevi

■ Aşağıdaki konulardan birini araştırmak üzere seçin:

- İş gereksinimleri analizi (*business requirements analysis*)
- Yazılım gereksinimleri analizi (*software requirements analysis*)
- Yazılım test (*software testing*)
- Yazılım doğrulama (*software verification*)
- Yazılım onaylama (*software validation*)
- Yazılım büyüklük kestirimi (*software size estimation*)
- Yazılım maliyet kestirimi (*software cost estimation*)
- Yazılım ölçme (*software measurement*)
- Yazılım süreç değerlendirme (*software process assessment*)
- Yazılım süreç iyileştirme (*software process improvement*)

(Bunlar dışında konu önerileri de getirebilirsiniz.)

A.Tarhan, 2010

- 59 -

BBS-651-DN05

2. Arasınan Ödevi (..devam)

■ Araştırmanızda hem konuyu detaylıca incelemeniz hem de konu kapsamında yer alan model, yöntem veya teknikleri önceden belirlenen özelliklere göre karşılaştırmanız beklenmektedir.

- Karşılaştırmayı hangi bağlamda ve hangi özelliklere göre yapacağınız konusunda, dersin eğitmeni tarafından geribildirim verilecektir.

■ Çıktı: Araştırma makalesi (giriş, inceleme, karşılaştırma, sonuç ve referansları içeren)

► Adım-1: (Teslim tarihi: 7 Aralık 2010 24:00; Teslim şekli: elektronik)

- Ön rapor teslimi (metin içinde yararlanılan referanslara atıflar yapılmalıdır.)
 - Konunun detaylı incelemesi
 - Konu kapsamında yer alan model, yöntem veya tekniklerin tanımı
 - Karşılaştırma bağlamı ve özellikleri

► Adım-2: (Teslim tarihi: 9 Ocak 2011 24:00; Teslim şekli: elektronik ve basılı kopya)

- Makale raporunun teslimi
 - Adım-1 kapsamında teslim edilen metne ek olarak;
 - Konu kapsamında yer alan model, yöntem veya tekniklerin, onaylanan bağlam ve özelliklere göre karşılaştırması ve
 - Sonuçlar

A.Tarhan, 2010

- 60 -

BBS-651-DN05