YMH311-Yazılım Tasarım ve Mimarisi
Mimari Tasarım

Prof. Dr. Resul DAŞ

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bu Haftaki Konular

Amaçlar

Mimari tasarım, ürün tasarımı ve ayrıntılı tasarım arasındaki ilişkileri tartışmak

Mimari tasarıma etki eden faktörleri listelemek

Mimari tasarım sürecini gözden geçirmek

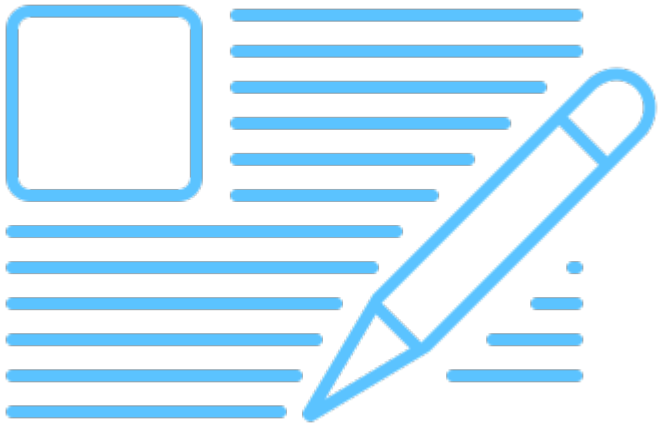
Mimari Tasarım Dokümanı'nın (SAD) içeriğini sunmak

Kalite niteliklerini ve bunların mimari tasarımdaki rolünü açıklamak

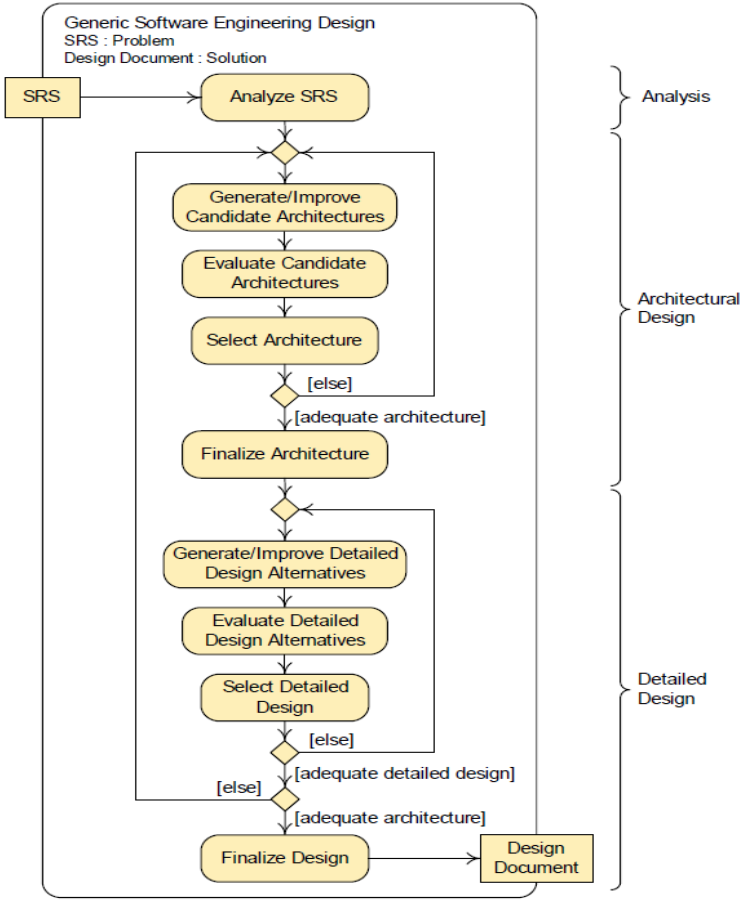
Mimari tasarım spesifikasyonu notasyonlarını ve özellikle arabirimler (interface) için olanlarını araştırmak

İçerik

- Mimari tasarım, ürün tasarımı, ve ayrıntılı tasarım
- Mimari tasarımı etkileyen faktörler
- Mimari tasarım süreci
- Kalite nitelikleri
- Mimari tasarım spesifikasyonları



Genel Bir Yazılım Mühendislik Tasarımı Süreci



Mimari Tasarım (Architectural Design)

Mimari tasarım, programın büyük parçalarının, bunların sorumluluklarının, özelliklerinin, ve arabirimlerinin; ve bu parçalar arasındaki ilişki ve etkileşimlerin belirlenmesi aktivitesidir.

Ürün Tasarımında Mimari Tasarım

Mimariye ürün tasarımı sırasında da gereksinim duyulur.

- Fizibiliteyi değerlendirmek için
- Paydaşları (stakeholders) gereksinimlerinin karşılanabileceğine ikna etmek için
- Fayda-maliyet analizi yapmak için
- Projeyi planlamak için

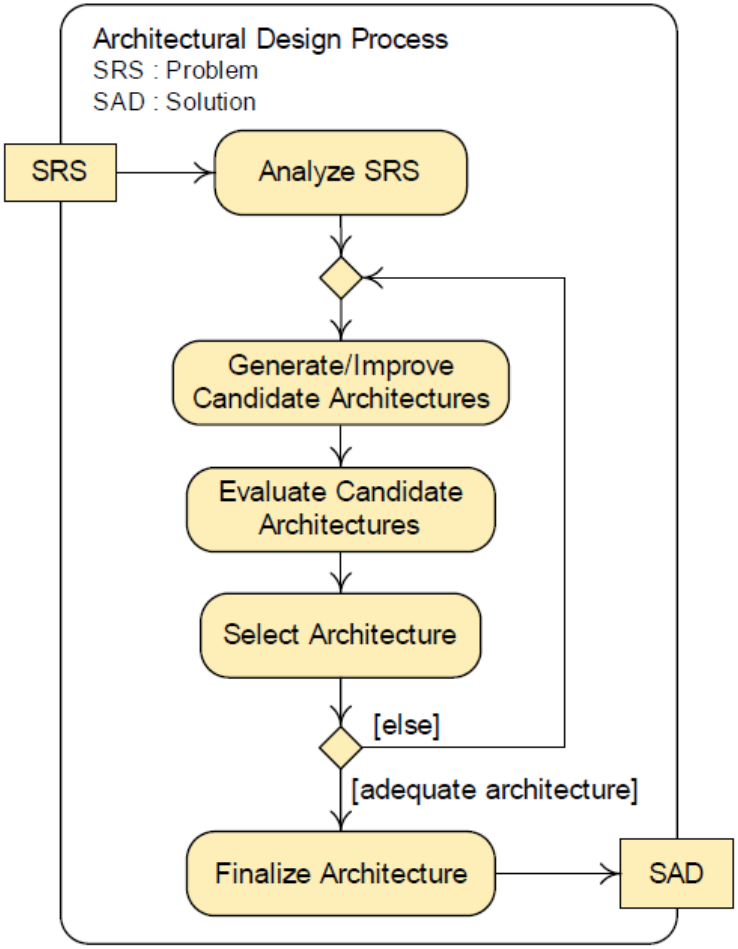
Mimari Tasarım ve Ayrıntılı Tasarım

- Mimari tasarım ve ayrıntılı tasarım arasındaki ayrım bazen yeterince belirgin olmayabilir.
  - “Büyük” bir program parçası ne demek?
  - Mimari spesifikasyonları ne kadar soyut (abstract) olmalı?
  - Çok küçük bir programın mimarisi nedir/nasıldır?

Mimari Tasarıma Etki Eden Faktörler

- Kod kütüphaneleri ve diğer kullanılabilir varlıklar (assets)
- Kurumsal yapı (Organizational structure)
- Tasarımcıların bilgi ve deneyimi
- Mimariler de kişileri ve kurumları etkileyebilir.

Mimari Tasarım Süreci



Mimari Tasarım Dokümanı (SAD)

- **Ürüne Genel Bakış (Product Overview)**—Ürün vizyonu, paydaşlar, hedeflenen pazar, vs.
- **Mimari Modeller (Architectural Models)**—Statik ve dinamik çeşitli modellere ilişkin spesifikasyonlar,
  - DeSCRIPTR
- **Modellerin Nasıl Eşleştiği (Mapping Between Models)**—Modellere ilişkin tablolar ve yazılı bilgiler
- **Mimari Tasarımın Gerekçesi (Architectural Design Rationale)**—Zor, çok önemli, kafa karıştıcı, ve değiştirilmesi zor tasarım kararlarına dair açıklamalar

Kalite Nitelikleri (Quality Attributes)

Bir **kalite niteliği**, müşteri gereksinimlerinin karşılanması bakımından önemli olan işlevlerinden bağımsız olarak, bir yazılım ürününün bir karakteristiği veya özelliğidir.

- İşlevsel olmayan gereksinimler
- Mimarilerin kalite nitelikleri üzerinde büyük etkisi vardır
- Geliştirmeye veya operasyona yönelik nitelikler
- Kalite niteliklerini kullanmak için teknikler (daha sonra)

Geliştirme Nitelikleri (Development Attributes)

- **Bakım Yapılabilirlik (Maintainability)**—Ürünün hatalarının düzeltilebilme, iyileştirilebilme ve başka platforma taşınabilme (port) kolaylığı,

  Genellikle altbölümlere ayrılır.

- **Yeniden Kullanılabilirlik (Reusability)**—Ürünün parçalarının başka ürünlerin geliştirilmesinde kullanılabilirliğinin derecesi

- Diğer

Operasyonel Nitelikler (Operational Attributes)

- **Performans (Performance)**—Ürün işlevlerinin zaman ve kaynak limitleri içerisinde sağlanabilme becerisi
- **Uygunluk (Availability)**—Kullanıma hazır bulunma durumu
- **Güvenilirlik (Reliability)**—Normal çalışma koşullarında gereksinimlere uygun davranabilme becerisi
- **Güvenlik (Security)**—Kötü niyetli uygulamalar ve etkiler karşısında zarar görmeye ya da zarar vermeye karşı koyabilme becerisi
- Diğer

Mimari Spesifikasyon Notasyonları

Spesifikasyon Türü	**Kullanışlı Notasyonlar**
Decomposition	Box-and-line diagrams, class diagrams, package diagrams, component diagrams, deployment diagrams
States	State diagrams
Collaborations	Sequence and communication diagrams, activity diagrams, box-and-line diagrams, use case models
Responsibilities	Text, box-and-line diagrams, class diagrams
Interfaces	Text, class diagrams
Properties	Text
Transitions	State diagrams
Relationships	Box-and-line diagrams, component diagrams, class diagrams, deployment diagrams, text

Arabirimler (Interfaces)

Bir **arabirim** varlıklar arasındaki iletişim sınırıdır.

Bir **arabirim spesifikasyonu**, bir varlığın içinde bulunduğu ortam ile haberleşmek için kullandığı mekanizmayı tanımlar.

Arabirim Spesifikasyonları

- Sözdizim (Syntax)—İletişim ortamının elemanları ve bunların mesaj oluşturmak için nasıl kombine edildiği
- Semantik (Semantics)—Mesajların anlamları
- Pragmatik (Pragmatics)—Mesajların ilgili bağlam içerisinde görevleri yerine getirmek için nasıl kullanıldığı

Arabirim spesifikasyonları bir modül ve onun içinde bulunduğu ortamla yaptığı iletişimin sözdizim, semantik, ve pragmatiklerini içermelidir.

Arabirim Spesifikasyonu Şablonu

1. **Sağlanan Servisler (Services Provided)**

   Sağlanan her bir servis için belirtilmelidir:

   a) Sözdizim (Syntax)

   b) Semantik (Semantics)

   c) Pragmatik (Pragmatics)

2. **Gereksinilen Servisler (Services Required)**

   Gereksinilen her bir servis adıyla belirtilmelidir.

   Servis açıklaması da eklenebilir.

3. **Kullanım Kılavuzu (Usage Guide)**

4. **Tasarım Gerekçesi (Design Rationale)**

Semantik (Semantic) Spesifikasyonu

- Bir **önkoşul (precondition)** bir aktivite veya operasyonun başlangıcında sağlanması (doğru olması) gereken bir koşuldur.

- Bir **sonkoşul (postcondition)** bir aktivite veya operasyonun bitiminde sağlanması (doğru olması) gereken bir koşuldur.

- Önkoşullar ve sonkoşullar bir operasyon gerçekleştiğinde ne olması gerektiğini yani operasyonun semantiklerini belirtirler.

Mimari Modelleme Notasyonları

Amaçlar

Mimari modellemede kullanılan çeşitli notasyonları göstermek

- Box-and-line diagrams
- UML package diagrams
- UML component diagrams
- UML deployment diagrams

Ortak UML notasyonlarını göstermek

- Notes
- Constraints
- Properties
- Stereotypes

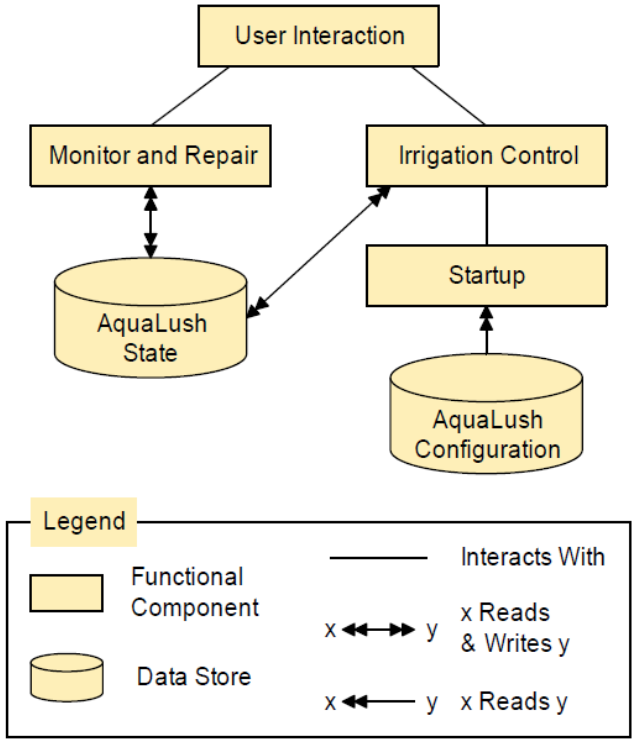
İçerik

- Kutu-ve-çizgi diyagramları (Box-and-line diagrams)
- Ortak UML notasyonları
- Paketler ve paket diyagramları (Packages and package diagrams)
- Bileşenler ve bileşen diyagramları (Components and component diagrams)
- Nodes, artifacts, and deployment diagrams

Kutu-ve-Çizgi Diyagramları (Box-and-Line)

- Çizgilerle birleştirilen simgeler (kutular, icon'lar) şeklindedir
- Oluşuma ilişkin kurallar yoktur
- Hem statik hem de dinamik modelleme için kullanılırlar
- Lejant (gösterge) eklenmesi iyi olur

Kutu-ve-Çizgi Diyagramı Örneği



Kutu-ve-Çizgi Diyagramı Kuralları

- Kutu-ve-çizgi diyagramlarını yalnızca standart notasyonların yetersiz kaldığı durumlarda kullanın.
- Kutuları ve çizgileri basit ve sade tutun.
- Farklı şeyler için farklı semboller/simgeler kullanın.
- Farklı diyagramlarda sembolleri tutarlı bir şekilde kullanın.
- Elemanları isimlendirirken gramer kurallarına uyun.
- Statik ve dinamik elemanları birlikte kullanmayın.

UML'de Notlar ve Kısıtlar

Not (Note)—Model elemanlarına kesik çizgiyle bağlanan, bir köşesi kıvrılmış bir kutu

- İsteğe bağlı herhangi bir metin içerebilir
- Yorumlar ve belirtimler için kullanılır

Kısıt (Constraint)—Model elemanları tarafından belirtilen varlıklar için sağlanması gereken koşulu belirten bir ifade

- Küme parantezi içinde yazılır { }
- Model elemanlarının yanında
- Çeşitli model elemanlarını bağlayan kesik çizgilerin yanında

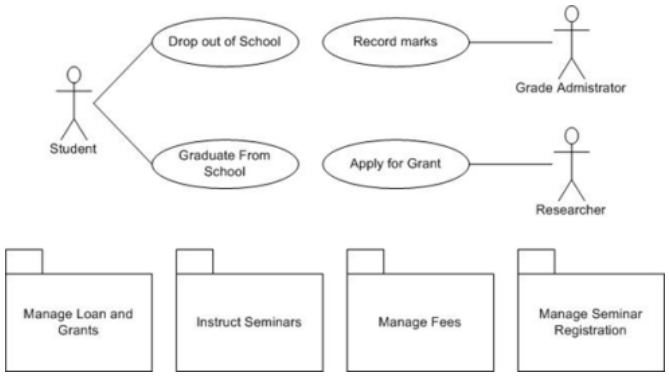
UML'de Özellikler ve Stereotipler

Özellik (Property)—Bir model elemanı tarafından belirtilen bir varlığın karakteristiği

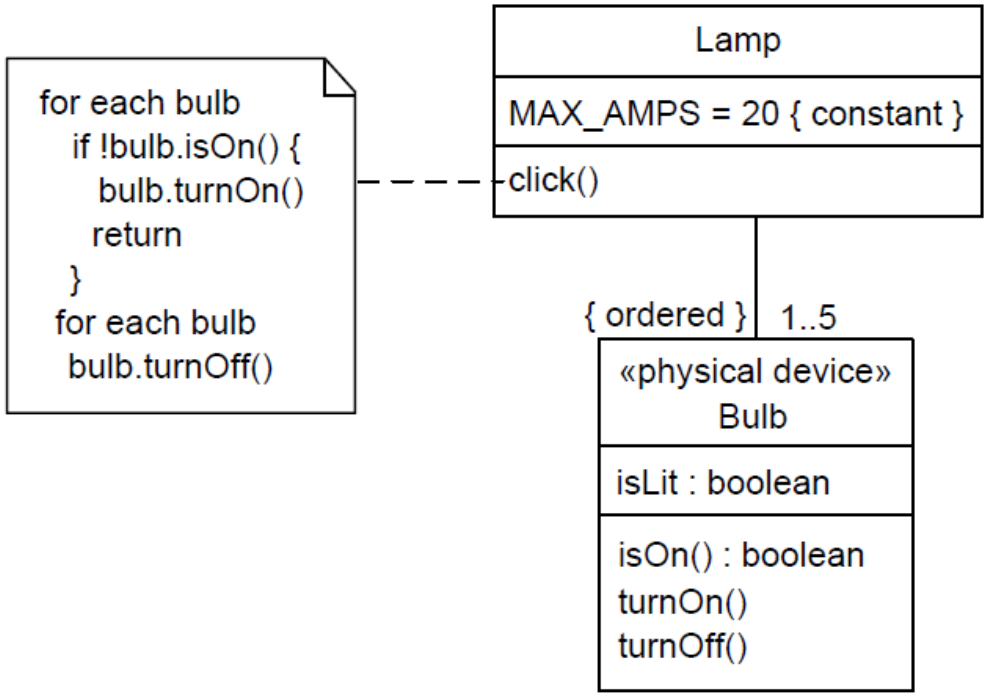
- Küme parantezi içinde etiketli değerlerin listesi
- Etiketli değer: etiket = değer
- True olan Boolean özellikler için değer ve eşittir simgesi yazılmayabilir.

Stereotip (Stereotype)—Daha spesifik anlam verilen bir model elemanı

- Simgeler, renkler, ve grafiklerle gösterilir
- Stereotip anahtar sözcükleri özel çift tırnaklar arasına yazılır, örneğin **«interface»**



Ortak Elemanlara Örnek

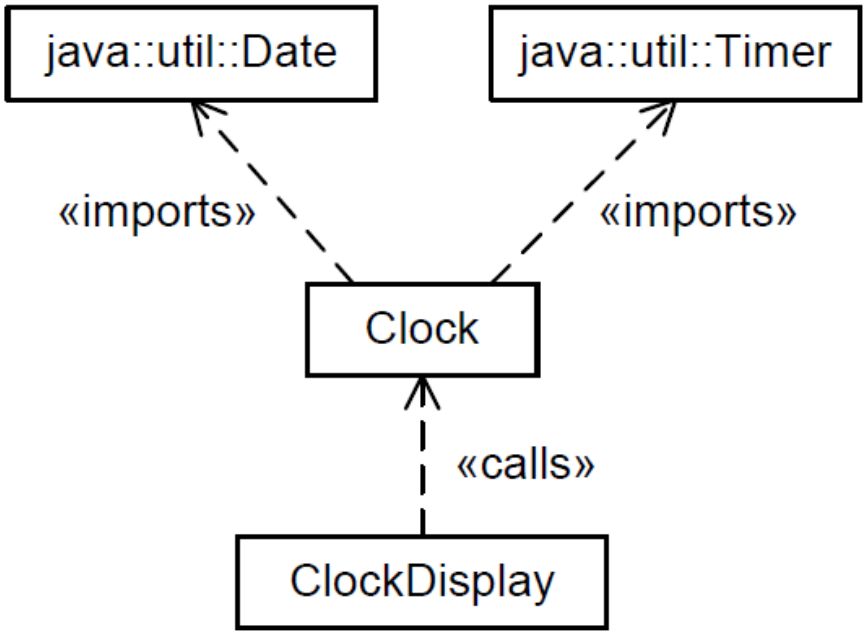


UML Bağımlılık (Dependency) İlişkileri

D ve *I* iki varlık olmak üzere; *I*'da (bağımsız - independent) olan bir değişiklik *D*'yi (bağımlı - dependent) etkiliyorsa, *D* ve *I* arasında bir **bağımlılık ilişkisi** (dependency relation) vardır.

- Örneğin: *D I*'yı kullanır, *D* derlenebilmek için *I*'ya bağımlıdır, *D I*'yı import eder.
- **Bağımlılık okları**yla (dependency arrow) temsil edilir: stereotipli kesik çizgili oklar.

Bağımlılık İlişkisine Örnek



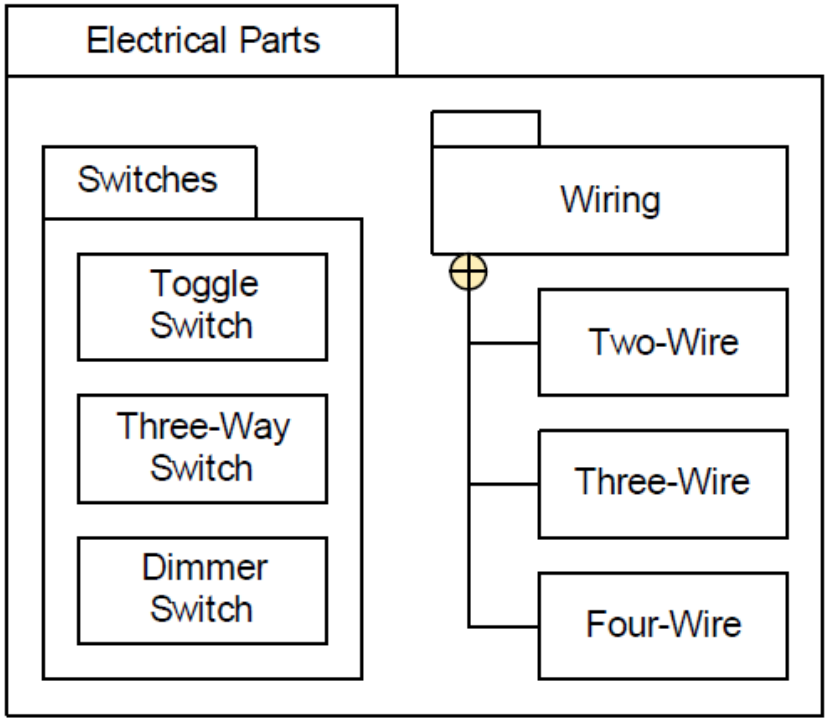
UML Paketleri (Packages)

- Bir UML paketi (package), paket üyeleri (package member) olarak adlandırılan model elemanlarının bir koleksiyonudur.
- Paket sembolu olarak dosya klasörü kullanılır.
  - Eğer gövde kısmı doluysa paket adı sekmeye, gövde dolu değilse gövde içinde değilse gövdeye yazılır
  - Üyeler gövde içinde ya da bir kapsama sembolü (çember içinde artı işareti) kullanılarak gösterilir
  - Genellikle import veya export bağımlılık oklarıyla bağlanırlar.

Paket Diyagramları

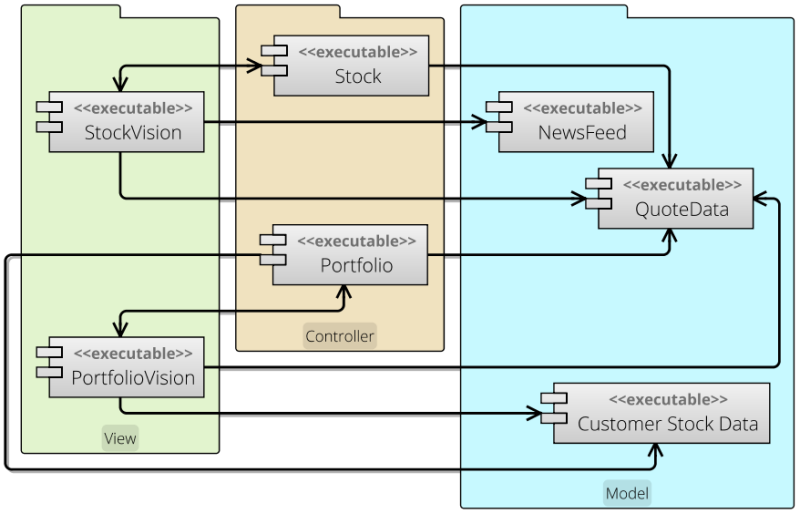
- Temel olarak paket sembolleri kullanılarak oluşturulan diyagramlara UML paket diyagramı denir.
- Kullanım alanı:
  - Modüllerin, parçalarının, ve onların ilişkilerinin statik modellerinin gösterimi
  - Mimari modelleme

Paket Diyagramı Örneği



Yazılım Bileşenleri (Components)

- Bir **yazılım bileşeni** yeniden kullanılabilir ve değiştirilebilir bir yazılım parçasıdır.
- **Bileşen-tabanlı yazılım geliştirme**, ürünlerin satınalınabilir ya da özel olarak geliştirilmiş yazılım bileşenleri kullanılarak tasarlandığı ve geliştirildiği bir yaklaşımdır.



UML Bileşen Diyagramları

- Bir UML bileşeni, iyi tanımlanmış arabirimlere sahip, modüler, ve başkasıyla değiştirilebilir (replacable) bir birimdir.
  - Bileşen sembolü isim içeren bir dikdörgendir
  - «component» ile stereotipli veya sağ üst köşesinde bileşen sembolü olabilir
- Bir UML bileşen diyagramı bileşenleri, ortamlarıyla ilişkilerini, ve iç yapılarını gösterir.

UML Arabirimleri (Interfaces)

Bir UML **arabirimi** (interface) public tanımlı özellikler ve soyut (abstract) operasyonların adlandırılmış bir koleksiyonudur.

- Stereotipli bir sınıf sembolüyle temsil edilir (daha sonra)
- Özel top ve soket sembolleriyle temsil edilir.

Not: arabirim (interface) sözcüğünün buradaki anlamı daha önce gördüğümüz iletişim sınırı anlamından farklıdır.

Sağlanan ve Gereksinilen Arabirimler

- Bir sınıf veya bileşen bir arabirimin tüm özelliklerini kendine dahil eder ve arabirimin tüm operasyonlarını implemente ederse, bu sınıf veya bileşen bu arabirimi **gerçekliyor/gerçekleştiriyor** (realize) demektir.

Sağlanan (Provided) arabirim—Sınıf veya bileşen tarafından gerçeklenir

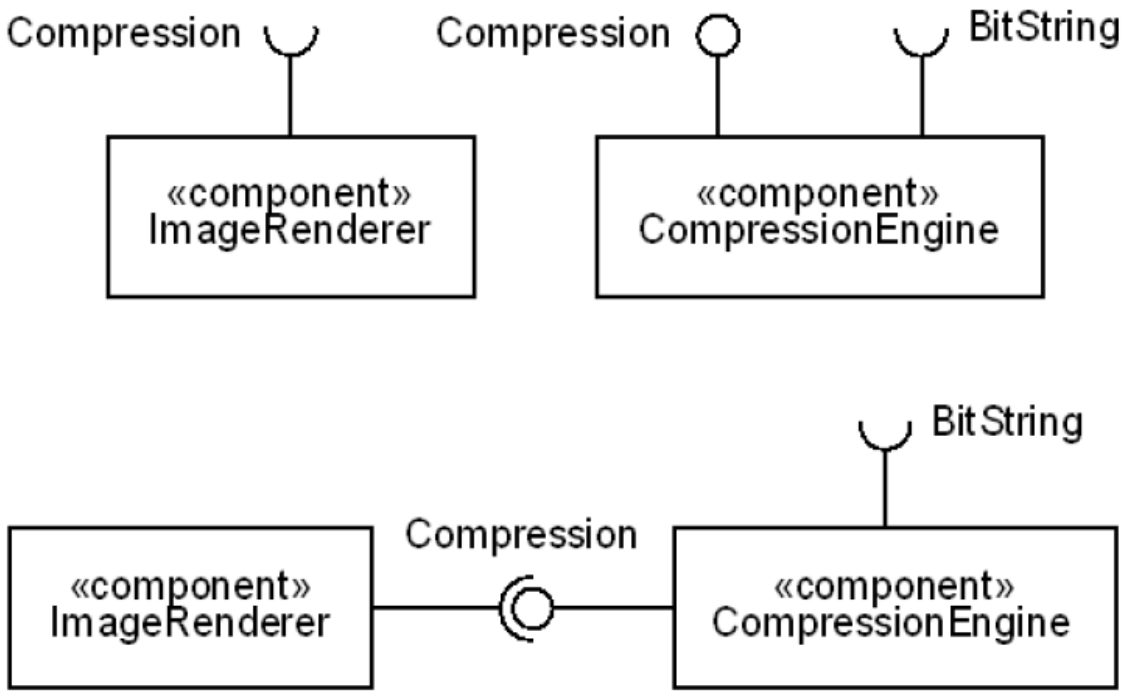
- Bir top veya lolipop sembolüyle temsil edilir

Gereksinilen (Required) arabirim—Bir sınıf veya bileşen tarafından ihtiyaç duyulur

- Bir soket sembolüyle temsil edilir

Montaj konnektörü (assembly connector) arabirimleri birbirine bağlar.

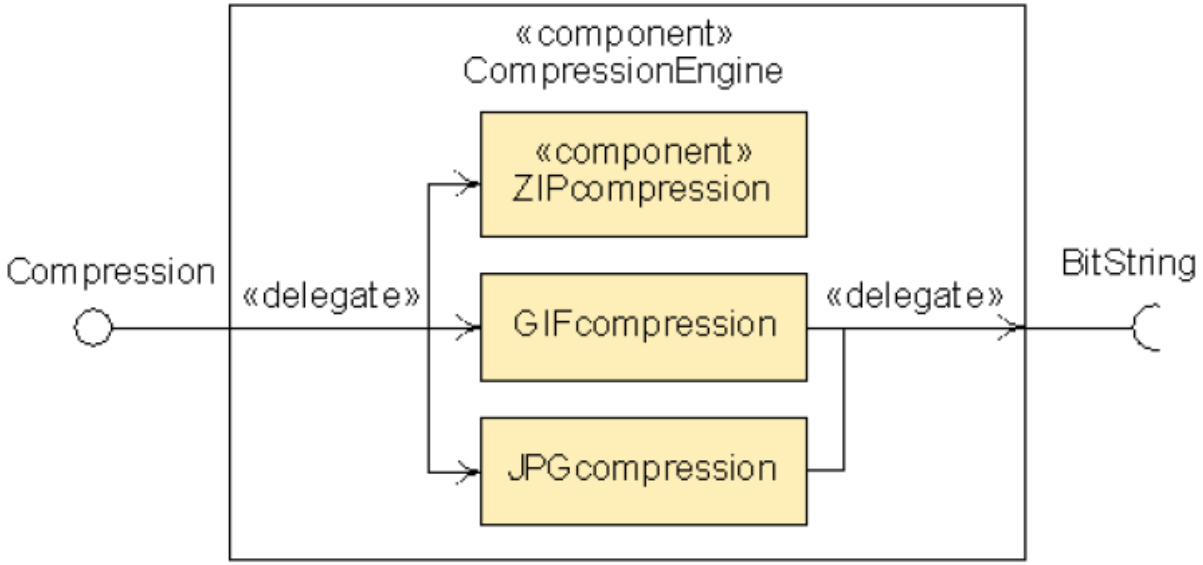
Arabirim Sembollerine Örnekler



Bileşenlerin İç Yapısı

- Bileşenler başka bileşenleri veya sınıfları içerebilirler ve içsel yapılarını gösterebilirler.
- Bir **delegasyon konnektörü** (delegation connector), bir bileşen arabirimini bu arabirimi gerçekleyen veya kullanan bir veya daha fazla içsel sınıfa veya bileşene bağlar.
  - Düz çizgili oklar
  - **«delegate»** ile stereotipli

Bileşenlerin İç Yapısına Örnek

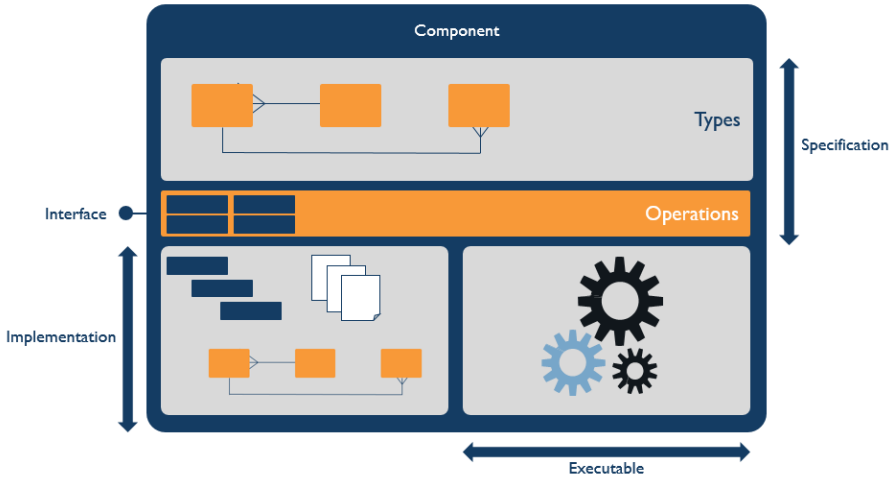


Bileşen Diyagramı Kullanımı

- Yazılım bileşenlerinin statik modellerinin gösteriminde (yeniden kullanılabilir ve başkasıyla değiştirilebilir parçalar)
- Program bileşenlerinin modellenmesinde
  - Mimari modeller
  - Ayrıntılı tasarım modelleri
  - Ortam ile ilişkiler
- Bileşenlerin içsel yapısının modellenmesinde kullanılır.

Mantıksal ve Fiziksel Mimari

- Mantıksal (Logical) mimari—Bir ürünün temel parçalarının ve onların ilişkilerinin çalışan kod halinde gerçek bir makinede implementasyonundan daha soyut bir konfigürasyonu
- Fiziksel (Physical) mimari—Bir ürünün işlemsel kaynaklar üzerinde kod ve veri dosyaları halinde bulunmasının ve çalışmasının gerçekleştirilmesi
- UML kurulum (deployment) diyagramı fiziksel mimariyi modeller.



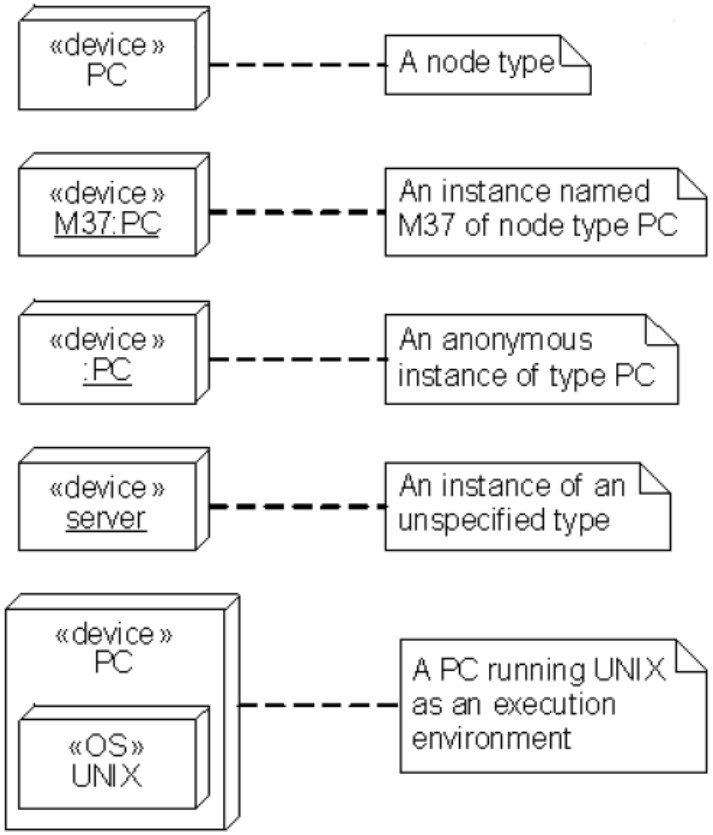
UML Artefaktları (Artifacts)

- Bir UML artefaktı (artifact) geliştirme veya işletim sırasında kullanılan veya üretilen verinin herhangi bir fiziksel temsilidir.
  - Örneğin: dosyalar, dokümanlar, program kodları, veritabanı tabloları, vb.
- Artefaktların tipleri ve örnekleri (instance) vardır
  - İsim içeren dikdörtgenle temsil edilir
  - **«artifact»** ile stereotiplidir veya sağ üst köşesinde artefakt sembolü vardır
  - Örneklerin isimlerinin altı çizgilidir, tiplerin isimlerini altı çizgili değildir
- Artefaktlar mantıksal varlıkların gerçeklenmiş şekilleridir (sınıflar, bileşenler, vb.)

UML Düğümleri (Nodes)

- Bir UML **düğüm**ü (node) işlemsel (computational) bir kaynaktır.
  - **Cihaz (Device)**—Bir bilgisayar gibi fiziklsel bir işlem birimidir
  - **İşletim ortamı (Execution environment)**—İşletim sistemi veya bir dil yorumlayıcısı gibi bir sanal makineyi implemente eden bir yazılım sistemidir
- UML'de kutu veya kütük (slab) sembolüyle temsil edilir
  - **«device»** veya **«execution environment»** ile stereotiplidir
  - Tipler ve örnekler (instance)
  - Tiplerin isimleri vardır
  - Örneklerin name : type formunda altı çizgili etiketleri vardır
  - İsim veya tipten biri yazılmayabilir, ama ikisi birden değil

Düğüm Sembolü Örnekleri



Kurulum (Deployment) Diyagramları

- Bir UML kurulum diyagramı, işlemsel kaynakları, aralarındaki iletişim yollarını, ve üzerlerinde bulunan ve işletilen artefaktları modeller.

Kullanım yeri:

- Sistemde kullanılan gerçek ve sanal makineleri göstermek
- Makineler arasındaki iletişim yolunu göstermek
- Sistemi oluşturan program ve veri dosyalarını göstermek
  - Yerleşim (Residence)
  - İşletim (Execution)

Kurulum Diyagramı Kuralları

İşlemsel kaynaklar düğümlerdir

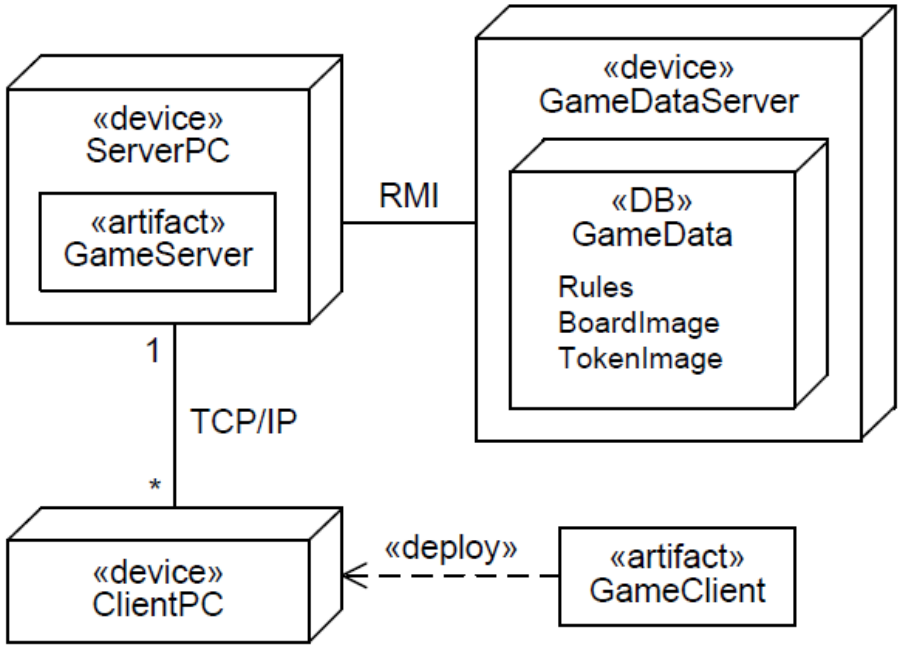
İletişim yolları düğümler arasındaki düz çizgilerdir

- Etiketlenebilir
- Çoklukları ve rol isimleri olabilir

Artefakt sembolleri

- Düğüm sembollerinin içinde görünebilir
- Düğüm sembollerinin içinde listelenebilir
- **«deploy»** stereotipli bağımlılık oklarıyla düğüm sembollerine bağlanabilir

Kurulum Diyagramı Örneği



Özet

- Mimari tasarım ürünün fizibilitesinin değerlendirilebilmesi amacıyla genellikle ürün tasarımı sırasında başlamalıdır.
- Mimari tasarımdan ayrıntılı tasarıma doğru soyutlamanın (abstraction) derecesi azalır.
- Bir mimarı tasarım dokümanı (SAD) ürüne genel bir bakış, mimari spesifikasyonlar, ve tasarım gerekçesi gibi bölümlerden oluşur.
- Mimari kararların kalite nitelikleri üzerinde büyük etkisi vardır.
- Mimari modelleme için kullanılan çeşitli notasyonlar mevcuttur.
- Arabirim spesifikasyonları, iletişim ortamına dair tanımlamalar içermelidir
  - Sözdizim (Syntax),
  - Semantik (Semantics), ve
  - Pragmatik.

Özet

- Kutu-ve-çizgi diyagramları statik ve dinamik mimari model oluşturmak için kullanılır.
- Notlar, kısıtlar, özellikler, ve stereotipler herhangi bir UML diyagramında kullanılabilir.
- Paket diyagramları modulleri ve modüllerin parçalarını modellemek için kullanılır.
- Bileşen diyagramları yazılım bileşenlerini modellemek için kullanılır.
- Kurulum diyagramları fiziksel mimarileri modellemek için kullanılır.

Kaynaklar

“Software Engineering A Practitioner’s Approach” (7th. Ed.), Roger S. Pressman, 2013.

“Software Engineering” (8th. Ed.), Ian Sommerville, 2007.

“Guide to the Software Engineering Body of Knowledge”, 2004.

” Yazılım Mühendisliğine Giriş”, TBİL-211, Dr. Ali Arifoğlu.

”Yazılım Mühendisliği” (2. Basım), Dr. M. Erhan Sarıdoğan, 2008, İstanbul: Papatya Yayıncılık.

Kalıpsiz, O., Buharalı, A., Biricik, G. (2005). Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme. İstanbul: Papatya Yayıncılık.

Buzluca, F. (2010) Yazılım Modelleme ve Tasarımı ders notları (http://www.buzluca.info/dersler.html)

Hacettepe Üniversitesi BBS-651, A. Tarhan, 2010.

Yazılım Proje Yönetimi, Yrd. Doç. Dr. Hacer KARACAN

YZM211 Yazılım Tasarımı – Yrd. Doç. Dr. Volkan TUNALI

http://www.cclub.metu.edu.tr/bergi_yeni/e-bergi/2008/Ekim/Cevik-Modelleme-ve-Cevik-Yazilim-Gelistirme

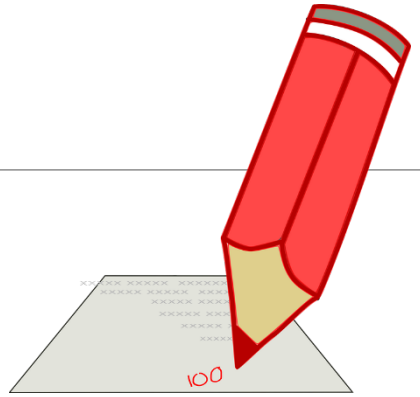
http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2011/ch6_6d_sk

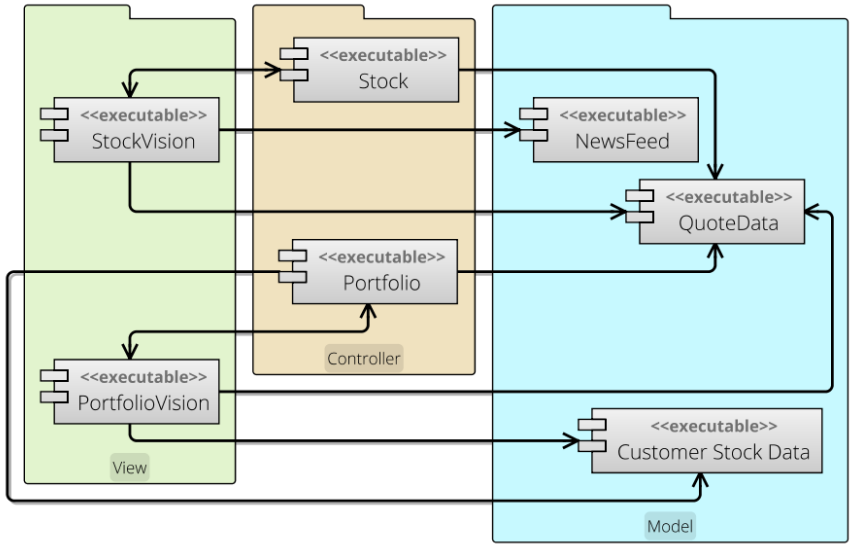
http://dsdmofagilemethodology.wikidot.com/

http://caglarkaya.piquestion.com/2014/07/01/244/

Ödev

Mimari Tasarım Hakkında Araştırma Yapınız.
Yazılım Mimarisinde Kullanılan Stilleri Araştırınız.





Sorularınız