

Konsistens mellan diagram och källkod

Data- och informationsvetenskap: Objektorienterad programmering och modellering för IA

Dagens agenda

- Modellen
- Tre nya diagramtyper
- Modellkonsistens
- Från diagram till kod och tillbaka

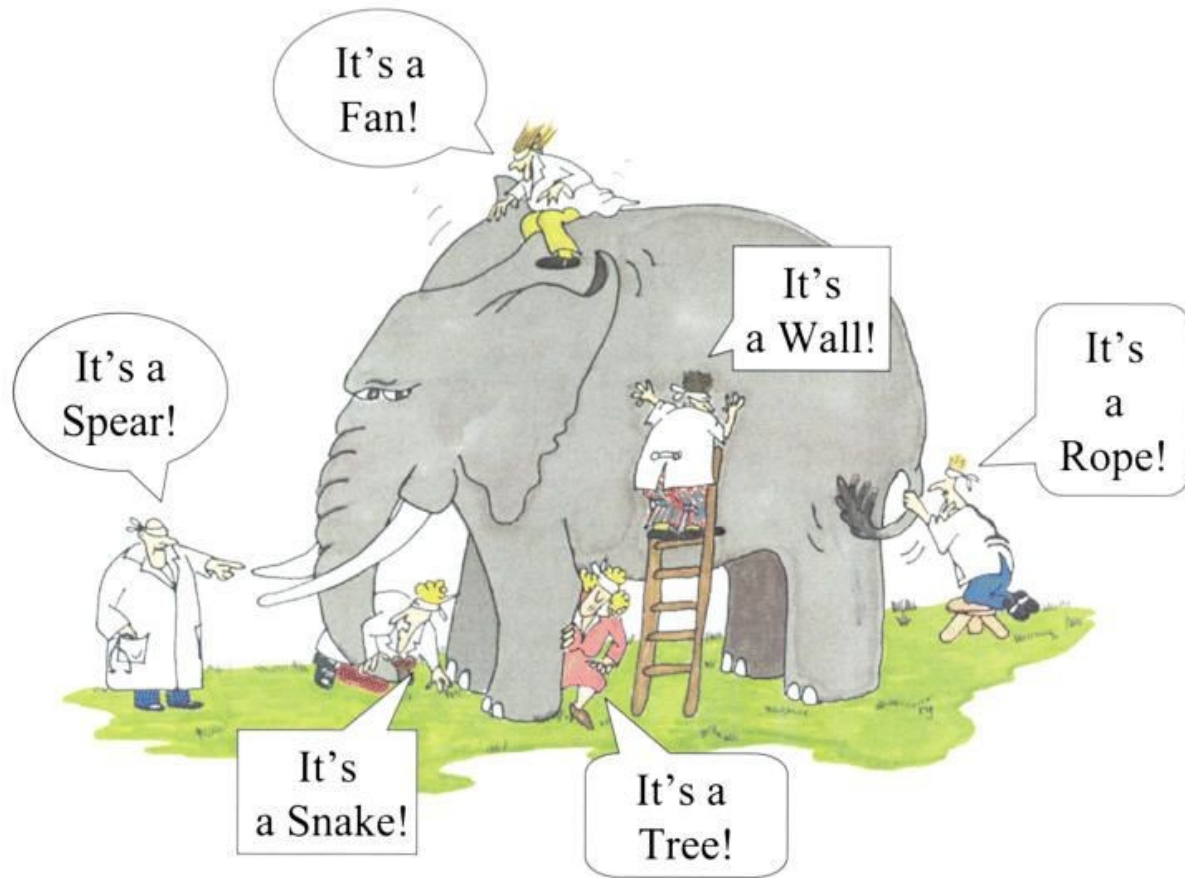
Modellen

Vad är en modell?

Modellen är en **bild**, eller en **tolkning**, av ett system.

Dess hela existensberättigande ligger i att utgöra en gemensam grund för utvecklarna – er – att gemensamt kunna diskutera och resonera kring.

Det finns många möjliga och giltiga tolkningar av ett system, vilket kan resultera i flera olika modeller.



The Blind Men and the Elephant

And so these men of Indostan
Disputed loud and long,
Each in his own opinion
Exceeding stiff and strong,
Though each was partly in the right,
And all were in the wrong!

John Godfrey Saxe

Testa själv!

En TV-apparat kan styras av en fjärrkontroll. Du som användare kan bland annat starta och stänga av TV:n, justera volymen och framförallt byta kanal. TV:n är en smart-tv, så den är i regel igång även som skärmen är avstängd. Den har en kanallista som du kan välja kanaler från. Kanalerna är antingen mappade mot en IP-ström (digital-tv) eller en frekvens (analog tv).

Hur använder vi modeller?

Som verktyg vid kravinsamling

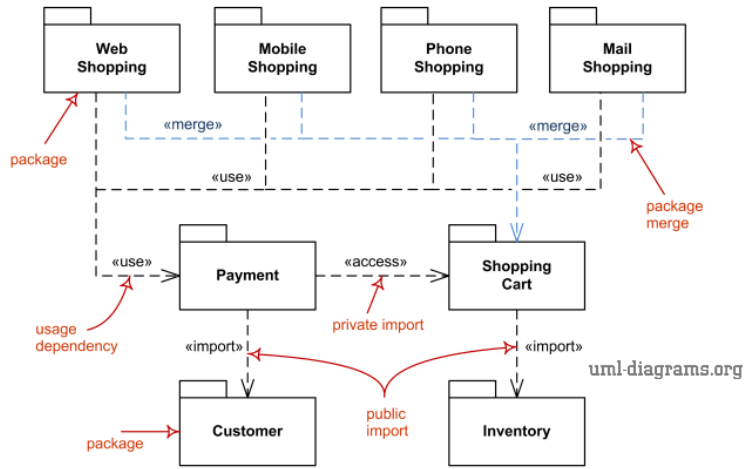
- Ett enklare sätt att kommunicera med kund/uppdragsgivare – tillsammans med UML!

Som verktyg vid implementation

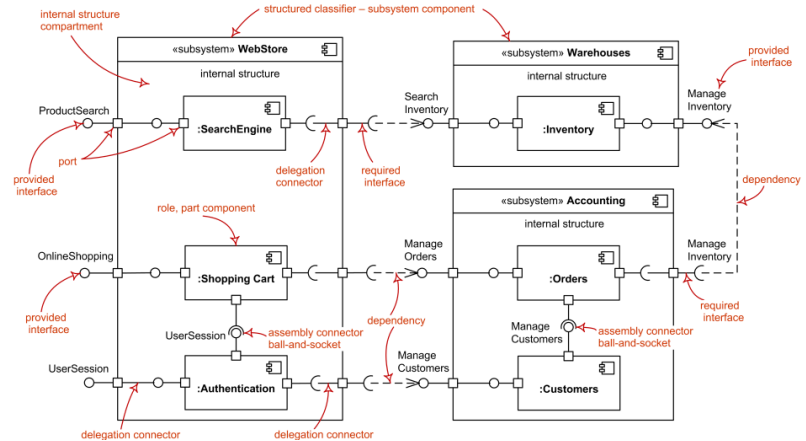
- Ett sätt för er som utvecklare att ha en gemensam bild av mjukvarusystemet
- Ett sätt att dela upp ett större system i mindre delar
- Tillsammans med tydliga och **konsistenta** UML-diagram: ett sätt att direkt generera kod!

Tre nya diagramtyper

Paketdiagram

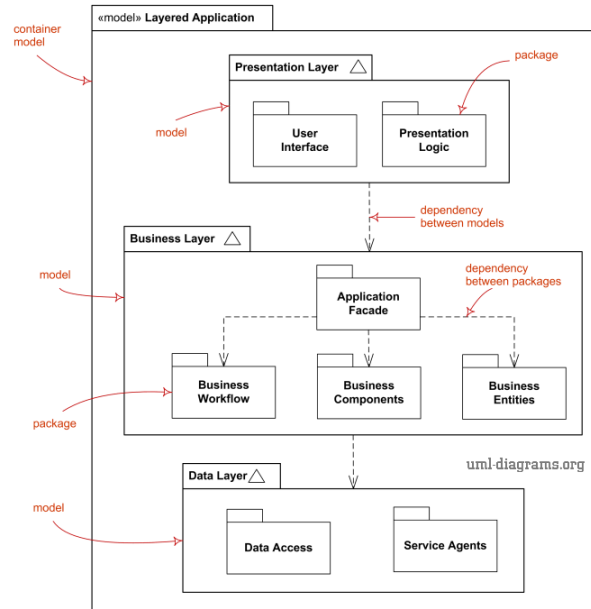


Komponentdiagram



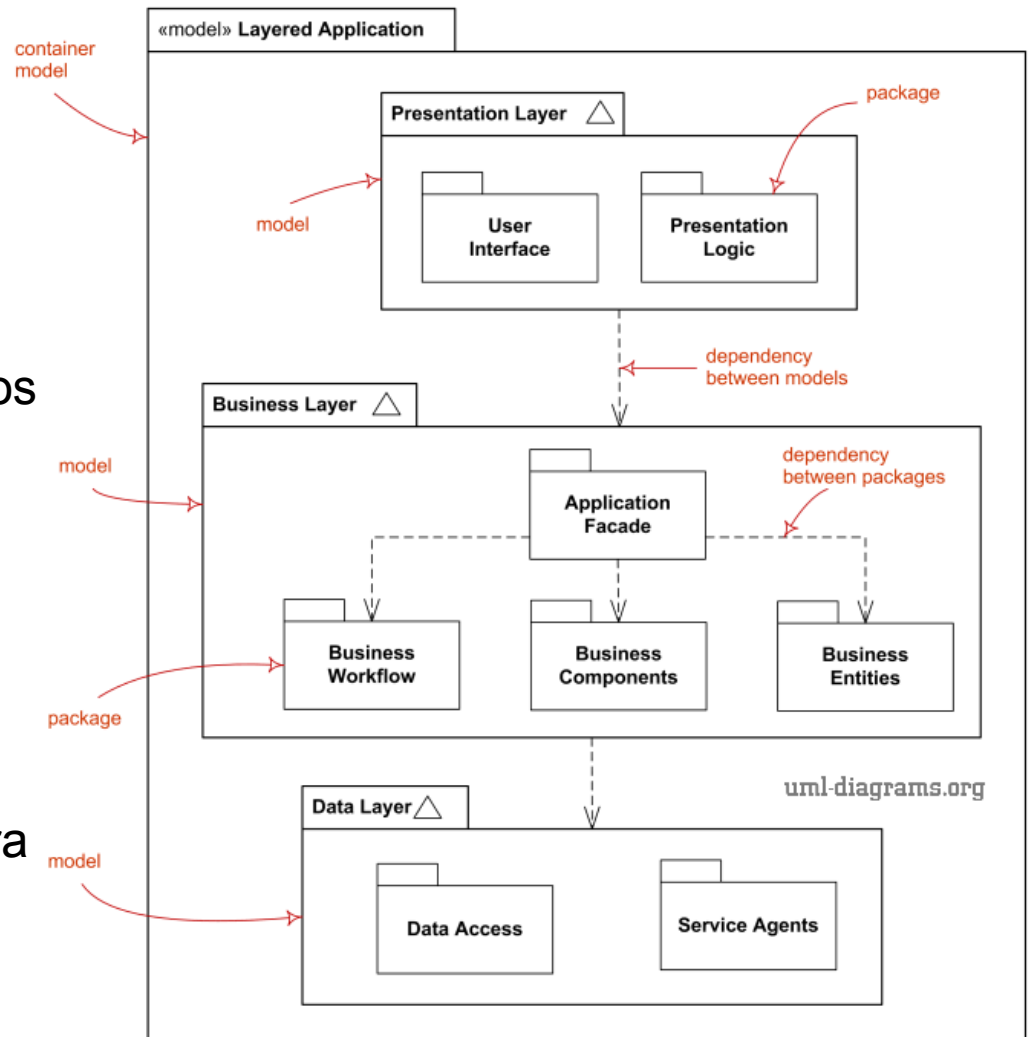
Tre nya diagramtyper

Modelldiagram



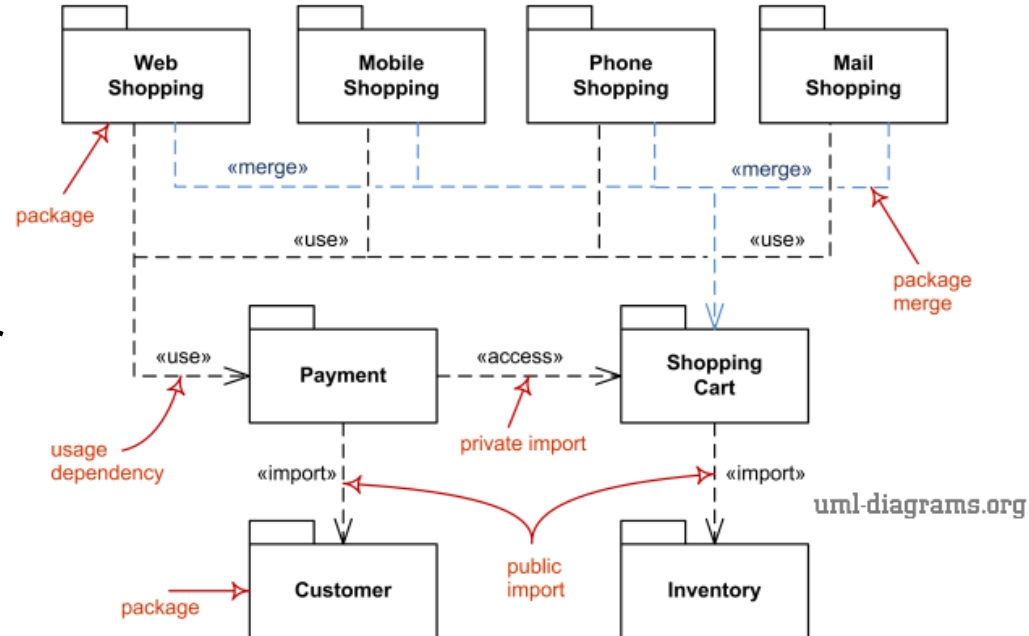
Modelldiagram

- Beskriver **strukturen** hos en modell
- Beskriver beroenden mellan **modeller** och **paket**
- Lagg märke till att en modell kan bestå av flera mindre modeller



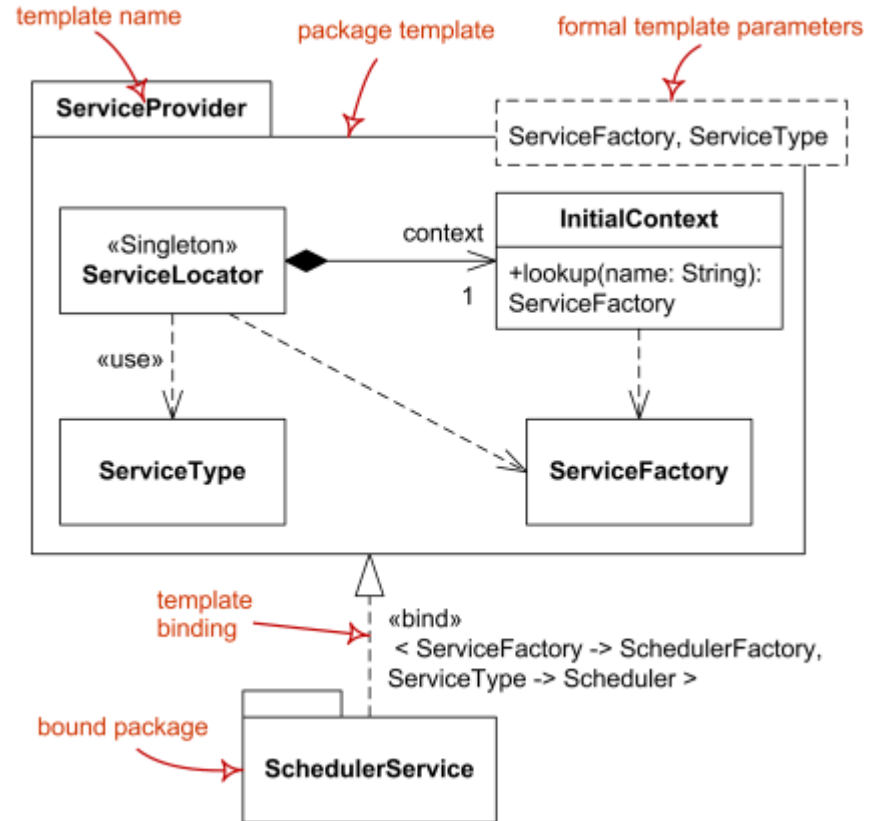
Paketdiagram

- Beskriver **paket** av klasser
- Bryter ner applikationer till mindre enheter – ät elefanten med skeden!



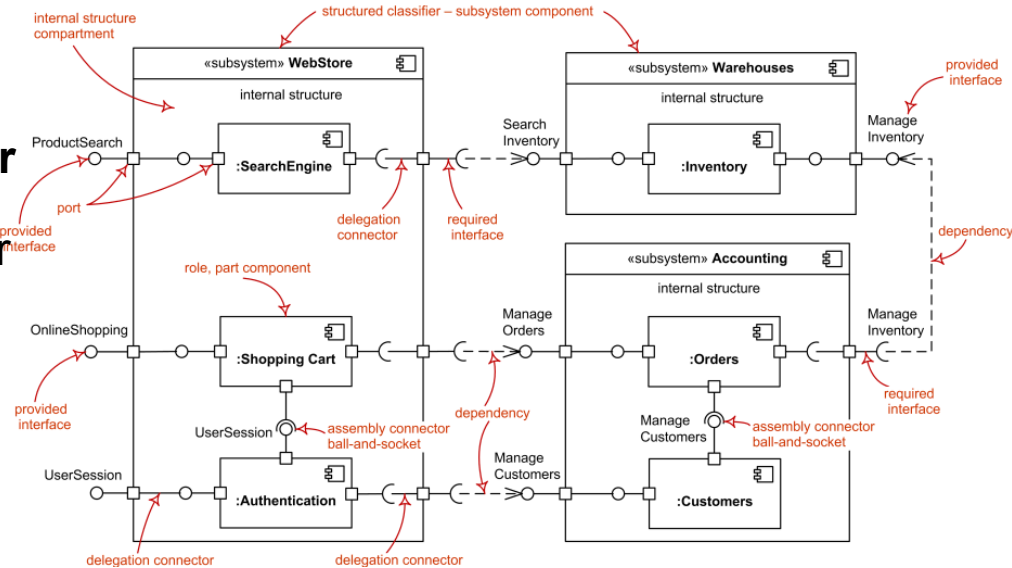
Paketdiagram

- Paketen beskrivs med vanliga klassdiagram



Komponentdiagram

- Beskriver applikationer som består av flera **komponenter**
- Dessa komponenter behöver inte nödvändigtvis köras på samma dator eller ens vara skrivna i samma språk
 - Det här tittar vi på i nästa läsperiod!



Vikten av en konsistent modell

Produktionsbortfall



Olyckor



Tappad försäljning

1 Biljettkaoset: Kunder 1 uppmanades åka gratis

54 Publicerad 6 okt 2017 kl 08:57



Eller värre...



Vad är konsistens?

Säkerställandet av att diagrammen i modellen faktiskt matchar varandra (och senare – källkoden)

Metoder för att säkerställa konsistens

Top – down och tillbaka

1. Börja med att analysera kraven/systemet
2. Identifiera användarfallen
3. Modellera användarfallen med aktivitetsdiagram
4. Bryt ner aktivitetsdiagrammen i sekvensdiagram
5. Gå uppåt i hierarkin och jämför syskonen i hierarkin
– matchar de varandra?

Mellan sekvens- och klassdiagram

1. Ta fram ett sekvensdiagram
2. Säkerställ att varje meddelande i diagrammet motsvaras av ett anrop till en existerande klass/objekt
 - Saknas något i klassdiagrammet för att kunna implementera en klass?
 1. Inför det i klassdiagrammet
 2. Kolla alla relaterade diagram så att de matchar förändringarna

Mellan klassdiagram och källkod

1. Ta fram ett klassdiagram
2. Översätt varje klass direkt från klassdiagrammet till källkod
 - Saknas något i klassdiagrammet för att kunna implementera en klass?
 1. Inför det i klassdiagrammet
 2. Kolla alla relaterade diagram så att de matchar förändringarna

Mellan källkod och klassdiagram

1. Ta en klass
2. Titta på hur klassen ser ut i modellen
 - Matchar de inte?
 1. Uppdatera klassdiagrammet
 2. Kolla alla relaterade diagram så att de matchar förändringarna

Mellan sekvensdiagram och källkod

1. Ta fram ett sekvensdiagram
2. Titta på hur sekvensen för ett specifikt objekt ser ut.
Gör anropen till omgivande klasser i samma ordning
som specificerats

Mellan källkod och sekvensdiagram

1. Ta fram ett sekvensdiagram
2. Titta på hur källkoden för den aktuella sekvensen ser ut.
 - Matchar de inte?
 - Uppdatera klassdiagrammet
 - Kolla alla relaterade diagram så att de matchar förändringarna

Slutligen

Läsanvisningar

Object-Oriented Systems Analysis and Design Using UML

- 5 Modeller
- 8.4 Komponenter
- 9.7-9.8 Modellkonsistens
- 11.7 Modellkonsistens
- 19.3 Komponentdiagram

UML consistency rules: a systematic mapping study

- <https://dl.acm.org/citation.cfm?id=2601292>