

# Användning av modeller för system/produktutveckling

Lars Wiktorin, IT plan  
lars.wiktorin@itplan.se

# Disposition

- Modellbegreppet
- Användningsområden
- Att välja modeller
  - Mottagare
  - Krav
- För system/produktutveckling
- Slutsatser

# Modellbegreppet

- Att skapa bilder av en företeelse ...
  - Avbildning – porträtt, karta, liknelse,
- En avsiktligt förenklad avbildning
  - Med avseende på vad?
    - Filter, perspektiv, utgångspunkt
  - Hur mycket förenklad?
    - Vad skall modellen förmedla?
  - Hur skall modellen utformas?
    - Beskrivningsspråk, formalism, ...

# Bilden är inte företeelsen

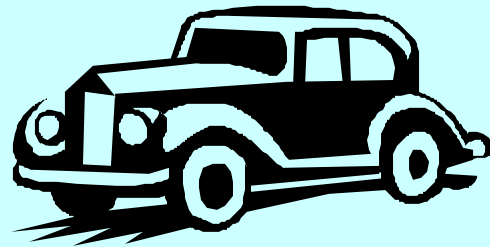


# Modell eller mall?

- Bilmodell

Mall

– eller



- Modellbil

Avbildning

”En modellbil kan vara en modell av en bilmodell”. Se upp med språket!

# Modelltyper

- En modell är av en viss typ
  - Jfr objekt och klass
- Vilka typer är relevanta?
  - Statisk - dynamisk
  - Formell - informell
  - Abstraherande nivåer
  - Perspektiv
- Kan en modell tillhöra flera typer?
  - Jfr multipelt arv inom OO

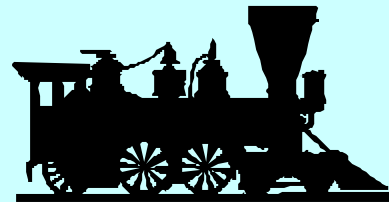
# Många är det - hur välja?

- En modell är av en viss typ
  - Jfr objekt och klass
- Vilka typer är relevanta?
  - Statisk - dynamisk
  - Formell - informell
  - Abstraherande nivåer
  - Perspektiv
- Kan en modell tillhöra flera typer?
  - Jfr multipelt arv inom OO

- Dynamisk, formell
  - $v(t) = \int a(t) dt$
- Informell begreppsmodell



- Informell statisk

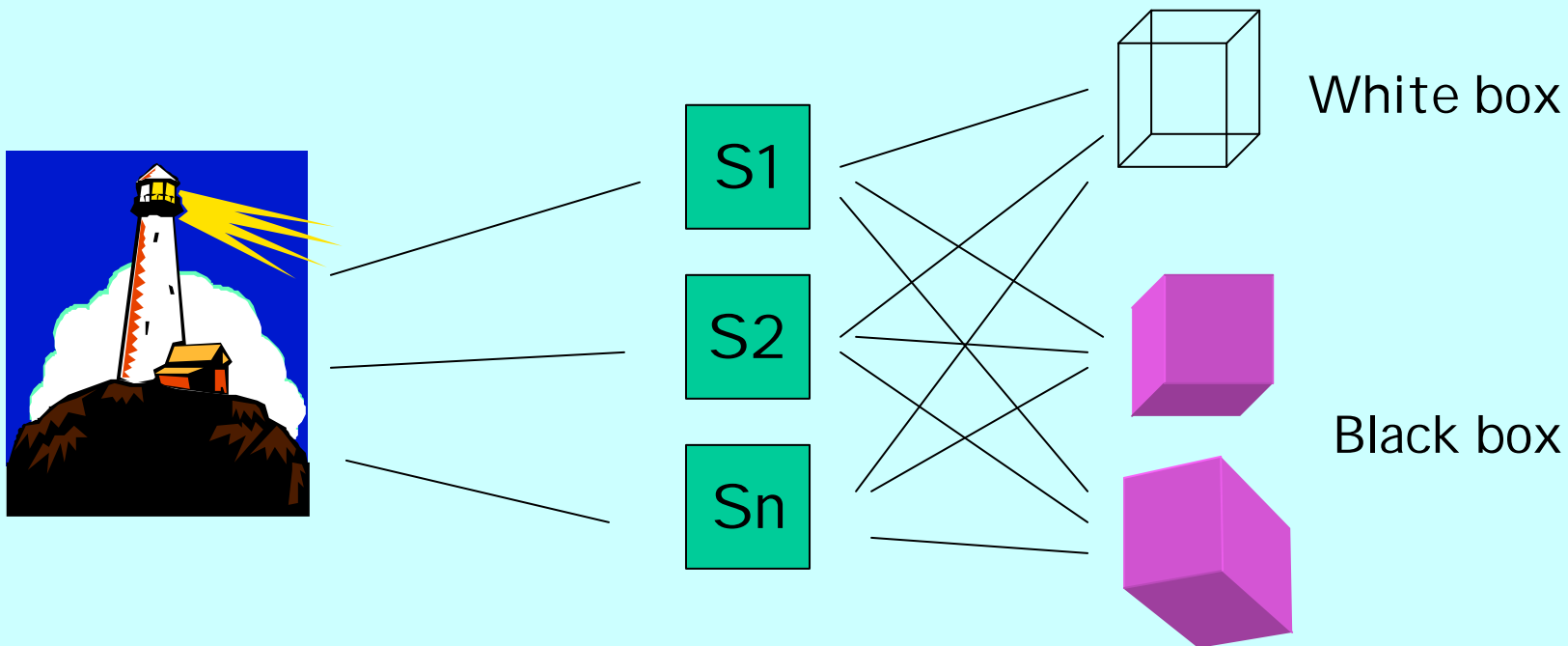


# Abstraktioner - Zachmans modell

	<i>Data, objekt Vad</i>	<i>Funktion, Uppgift Hur</i>	<i>Nätverk Var</i>	<i>Människor Vem</i>	<i>Tid När</i>	<i>Motiv Varför</i>
<i>Omfattning/mål</i>	Förteckning över nyckelbegrepp	Förteckning över processer	Förteckning över platser	Förteckning över organisationer	Förteckning över händelser	Strategi, mål, ...
<i>Verksamhetsmodell</i>	Konceptuell modell, VO	Processflöden	Logistiknät	Organisationsstruktur	Händelse-scheman	Affärsplan
<i>Informationssystemmodell</i>	Logisk modell, VO och gränssnitt	Uppgiftsmodell med script, workflow	Distribuerad systemark.	MDI arkitektur	Processstruktur	Kunskapsstruktur
<i>Teknikbegränsad modell</i>	Designmodell	Strukturdiagram, sekvensdiagram	Systemark	MDI design	Styrstruktur	Kunskapsdesign
<i>Detaljrepresentation</i>	Fysisk modell, typer, klasser	Programkod	Nätverks ark	Säkerhet, ...	Avbrott	Kunskapsbas
<i>System i drift</i>	Data	Funktion	Kommunikation	Organisation	Tidplan	Användning



# Perspektiv och utsiktspunkt



Utsiktspunkt  
Viewpoint

Är det skillnad  
på dessa?

Perspektiv  
Synvinkel

# Black/white box

## Black box

- Utifrån och in
- Gränssnitt
- Tjänster
- Aktörer
- Användningsfall

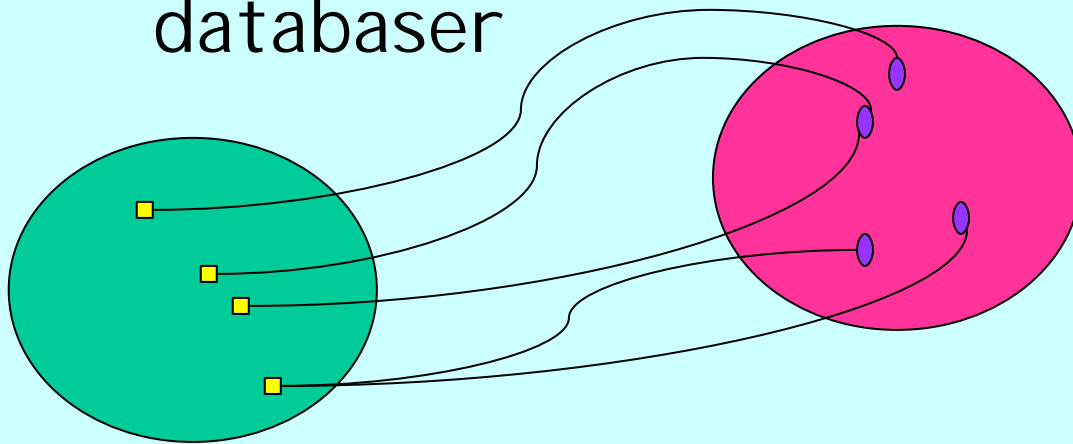
## White box

- Struktur
- Samband
- Flöden
- Algoritmer
- Data

Att gå från B->W är analys och design

# Avbildningar

- Eng. map och mapping (från -> till)
- Samband mellan modell och verklighet
  - Allt eller enbart delar, aggregeringar
  - Exempel: Relationsmodellen för databaser



Samband

1:1

1:N

N:1

N:M

# Användningsområden

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• Kommunlicera<ul style="list-style-type: none"><li>- Gemensam uppfattning</li></ul></li></ul> | <ul style="list-style-type: none"><li>• Beställare</li></ul>                  |
| <ul style="list-style-type: none"><li>• Analysera<ul style="list-style-type: none"><li>- Hur fungerar ...</li></ul></li></ul>        | <ul style="list-style-type: none"><li>• Utvecklare,<br/>reparatörer</li></ul> |
| <ul style="list-style-type: none"><li>• Instruera<ul style="list-style-type: none"><li>- Hur gör man för att ...</li></ul></li></ul> | <ul style="list-style-type: none"><li>• Användare</li></ul>                   |
| <ul style="list-style-type: none"><li>• Exekvera<ul style="list-style-type: none"><li>- Vad händer om ....</li></ul></li></ul>       | <ul style="list-style-type: none"><li>• Utvecklare,<br/>forskare,</li></ul>   |

# Att välja modell

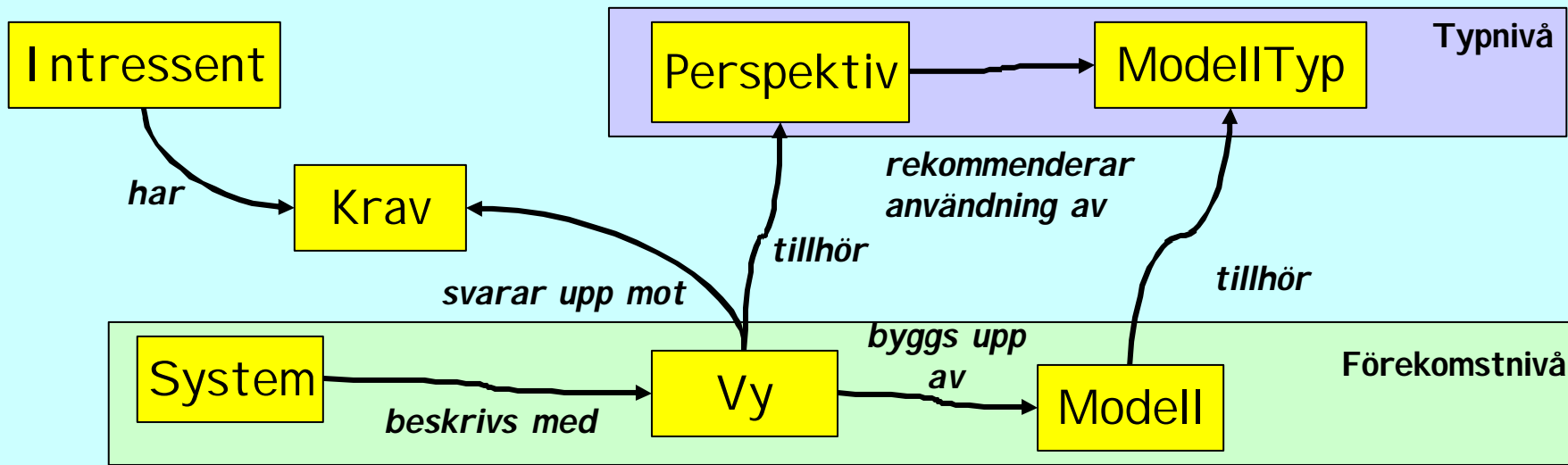
För vem?

Intressenterna

För att förklara vad?

Kraven

->Välj vyer som svarar mot kraven



Fritt efter IEEE1471

# Vy, perspektiv och modell

- Ett perspektiv rekommenderar ett antal modelltyper
- En vy följer ett perspektiv och innehåller ett antal modeller
- Exempel
  - Perspektiv: Funktion
  - Vyer: Klass, Användningsfall
  - Modelltyper: Statiska och dynamiska

Indelningen i exemplet kan diskuteras men avsikten borde vara klar

# Konsekvenser

- Olika intressenter kräver olika perspektiv
- Det kan löna sig att enas om ett antal grundperspektiv
- Exempel
  - Arkitekturramverk - DoD, IEEE1471

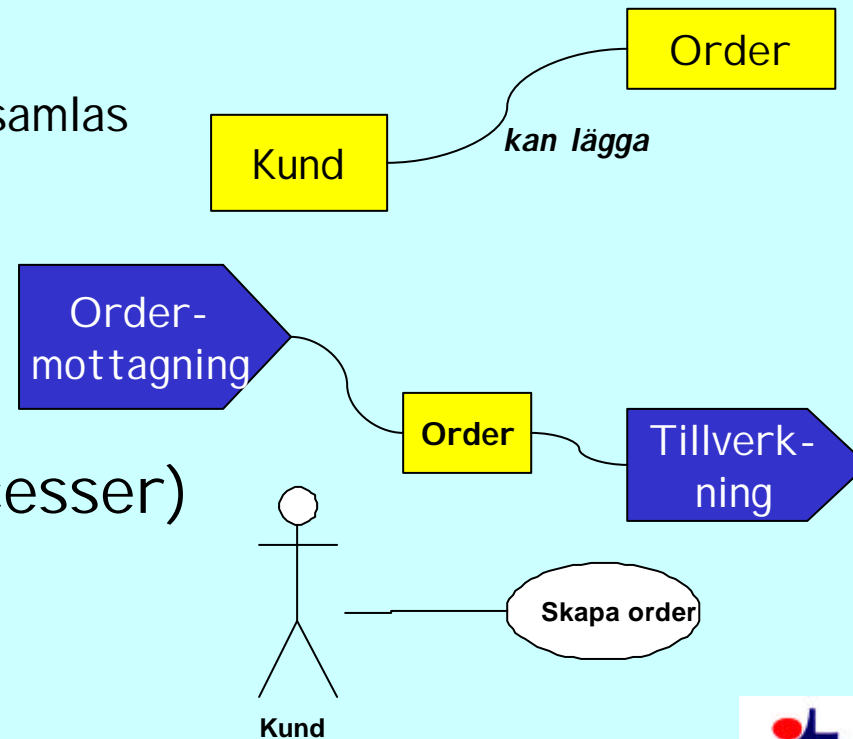
# Möjliga arkitekturperspektiv

- IEEE 1471
  - Enterprise, Information, Computation, Engineering, Technology
- FMA
  - System, Verksamhet, Organisation, Personal, Information, Teknik
- RUP
  - Logical, Implementation, Process, Deployment, Use Case .
- Business Architecture (Eriksson, Penker. See ref.)
  - Vision, Process, Structure, Behavior
- Herzum Business Component Factory
  - Functional, Application, Technical, (Proj mgmt)
- Siemens (Hofmeister, Appl SW Arch. See ref.)
  - Conceptual, Module, Code, Execution



# Verksamhetsmodellering

- En gruppverksamhet som avser att nå
  - gemensam förståelse, som samlas
  - i begripliga beskrivningar
- Modelltyper
  - Begreppsmodeller
  - Flödesmodeller (t ex processer)
  - Funktionsmodeller (t ex användningsfall)



# Objektmodeller(ing)

- Objekt eller objektorientering?
- Objektmodell
  - Trad beteckning på en dataorienterad statisk modell
- Objektorienterad modell
  - Tekniknära modell som fyller OO kraven
    - Inkapsling av beteende och egenskaper
    - Samverkan via meddelande utbyte
    - Klass och förekomst - arv mellan klasser

# Exekverbara modeller

- Simulering
  - Vad händer om?
- Styrning
  - Processindustri
  - Servomekanismer
- Förenklad konstruktion
  - Programvaruutveckling

# MDA - Model Driven Architecture

- Att skapa körbara system från **modeller** som definieras och förvaltas oberoende av exekveringsmiljö
- Tekniken förändras fortare än verksamheten
- Håller isär verksamhetsfunktioner och teknisk realisering
- Initiativ från OMG

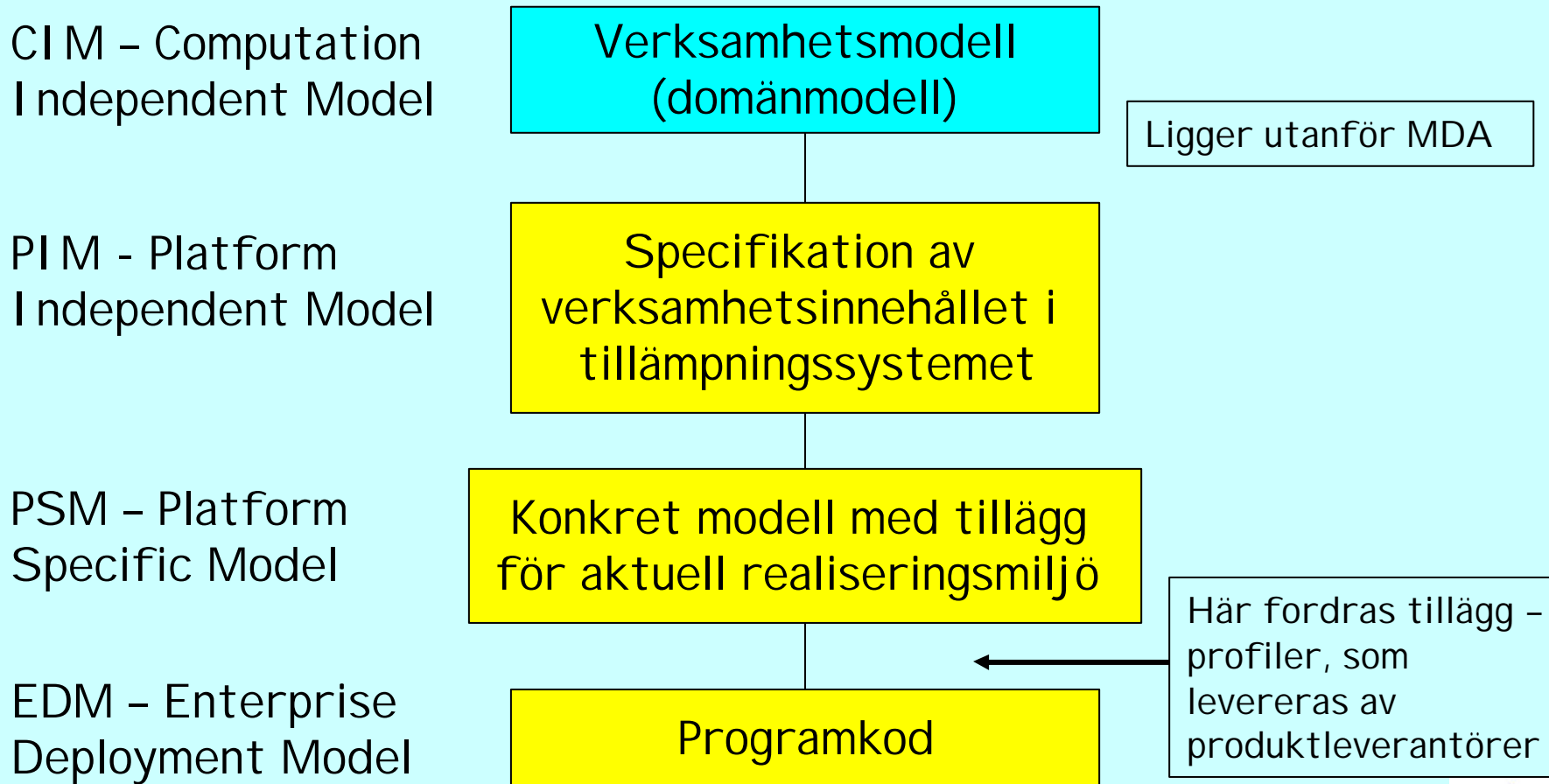
# Det här har vi hört förut

- Integrated CASE
  - Att skapa en miljö som hanterar modeller och överför dem till programkod
  - Ex COOL GEN, fd Texas Instruments IEF
- Repositories
  - Ex AD Cycle från IBM
- Fungerade inte bl a pga omogen och komplicerad teknik

# Vad är annorlunda denna gång?

- Ett gemensamt modellspråk – UML
- Etablerade modeller och specifikationer för infrastruktur
- Definierade API er till programspråk (profiler)
- Mer omfattande och användbara programmeringsmiljöer
- Gemensamt leverantörsintresse
  - 17 st nämns i nyhetsbrevet Comp Dev Strat

# MDA omfattar tre nivåer



# Kom ihåg

- En modell är **inte lika med** det avbildade systemet.
- En black box kan inte förväntas exekvera som om den vore en white box.
  - Det fordras tillägg
    - Regler, algoritmer, datavärden
  - Kan komma "från vänster" - externa leverantörer
    - jfr MDA PSM och PDM
  - Men vem tar fram PIM från CIM?



# MDA kräver programmering

- UML innehåller
  - Diagram klass, aktivitet, tillstånd, sekvens ,....
  - OCL - Object Constraint Language
- UML saknar
  - Möjlighet att beskriva innehållet i aktiviteter och åtgärder
  - Detta kommer i UML v2 - AL, Action Language
  - AL specificerat på semantisk men ej syntaktisk nivå

# Sammanfattning

- Modellen är inte systemet
- Modellen kan inte göra mer än vad den är avsedd för
- Att beskriva ett komplext system kräver många perspektiv och modeller
- Välj modell med hänsyn till syfte och mottagare