

# Návrh softwarových systémů - MDA

Martin Tomášek *Návrh softwarových systémů  
(B6B36NSS)*



```
public final void onSensorChanged(SensorEvent event)
{
    m_flightIntensity = event.values[0];
    m_etAmblight.setText("" + m_flightIntensity + " lx");
}
...
resume()
...
light, ... NORMAL);
```

# Co je to MDA?

- MDA = model driven architecture
- Založené na standardech OMG
- Hlavní myšlenkou je oddělit business a
- aplikační logiku od technologické platformy
- Přináší postupy, jak tyto modely správně transformovat.
- Primárními cíli tohoto přístupu jsou zajištění přenosnosti, interoperability (součinnosti) a znovupoužitelnosti díky oddělené architektuře.

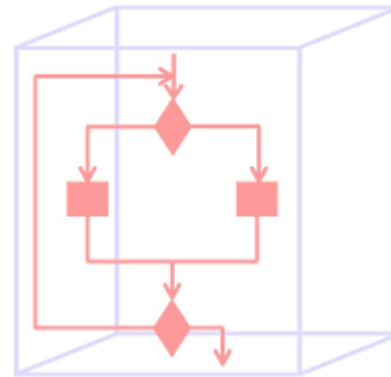


# MDA

- MDA = Model  $\rightarrow$  Code
- Model  $\leftarrow \rightarrow$  Code
- Existuje model i kód vedle sebe (náročné na udržovateľnosť)
- Code  $\rightarrow$  Model
- Kód je i modelom

# Typy modelů

- Datový
- Logický
- Konceptuální
- Objektový
- Entitně-vztahový
- Relační
- Ontologický
- Kontextový
- Funkční
- A další



# Důležité modely

- Computation Independent Model (CIM)

- výpočetně nezávislý model

- vytvořen z výpočetně nezávislého úhlu pohledu. Je určen pro analytika na nejvyšší úrovni. Měl by být vhodný i pro komunikaci se zákazníkem. Zásadně neřeší otázku, jak se co implementuje, jeho hlavní cíl je popsat co má systém dělat z pohledu zadavatele.

- **Requirements**

# Důležité modely

- Platform Independent Model (PIM)
- je na platformě nezávislý model
- platformě nezávislý úhel pohledu.
- Je nutné však podotknout důležitou skutečnost, že koncepce MDA hovoří o specifické míře nezávislosti na platformě. Jednoduše řečeno, model závislý na platformě obecného objektově orientovaného paradigmatu je však modelem na platformě nezávislým (PIM) vzhledem k množině konkrétních OO platforem (například C++, Corba, Smalltalk,...). Lze tak vlastně mluvit o celé hierarchii modelů PIM.
- **Design**

# Důležité modely

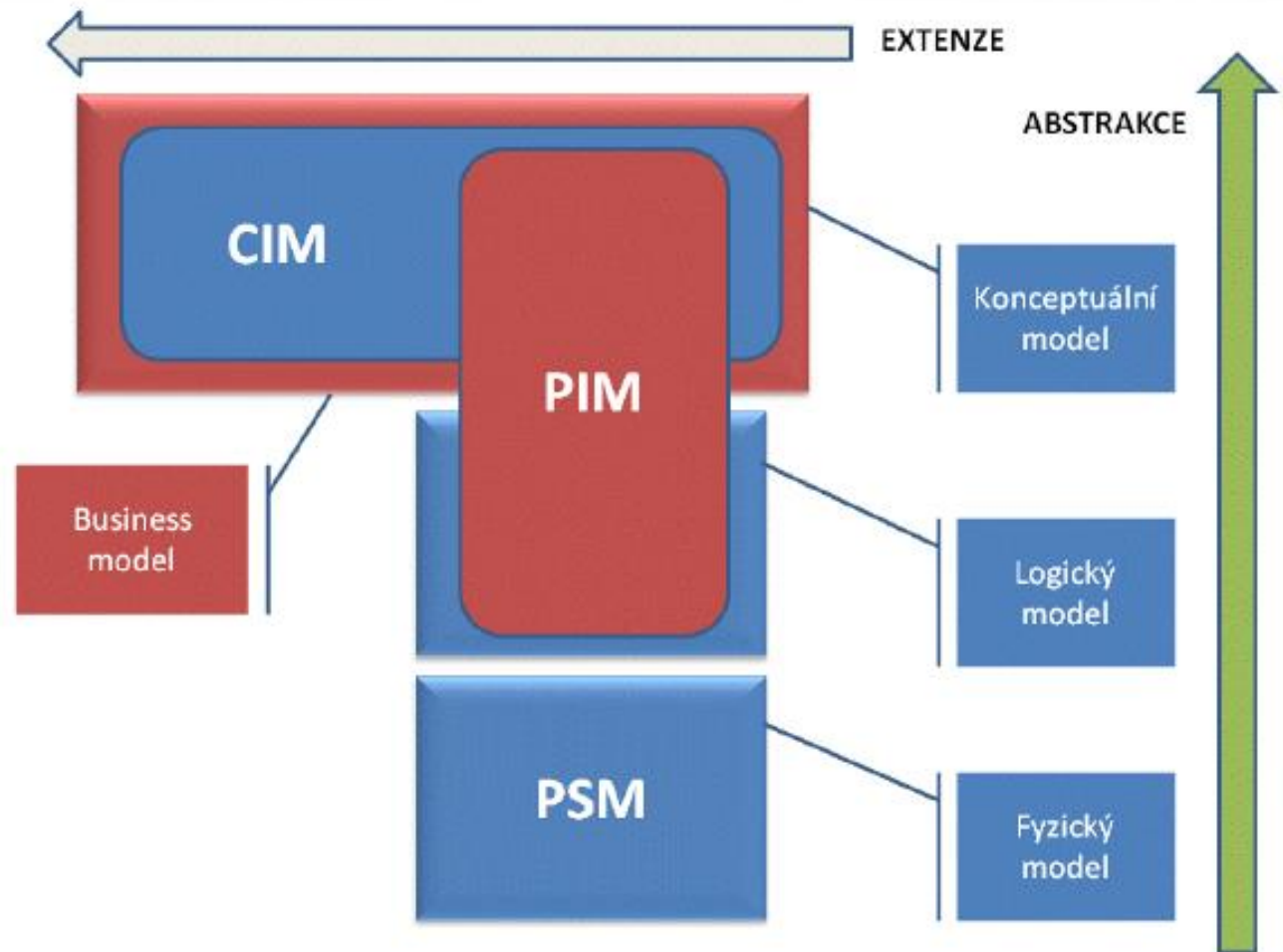
- Platform Specific Model (PSM)
- je na platformě závislý model
- Vychází z PIM a doplňuje jej o závislé detaily na platformě. Může existovat mnoho PSM pro různé platformy, které vycházejí z jediného PIM.
- **Implementation**

# Důležité modely

- Platform Description Model (PDM)
- je model popisující platformu
- Popisuje technologické koncepty určité platformy, které mohou být využity při realizace aplikace. Př. : CORBA Component Model.
- **Architecture design**



# Modely

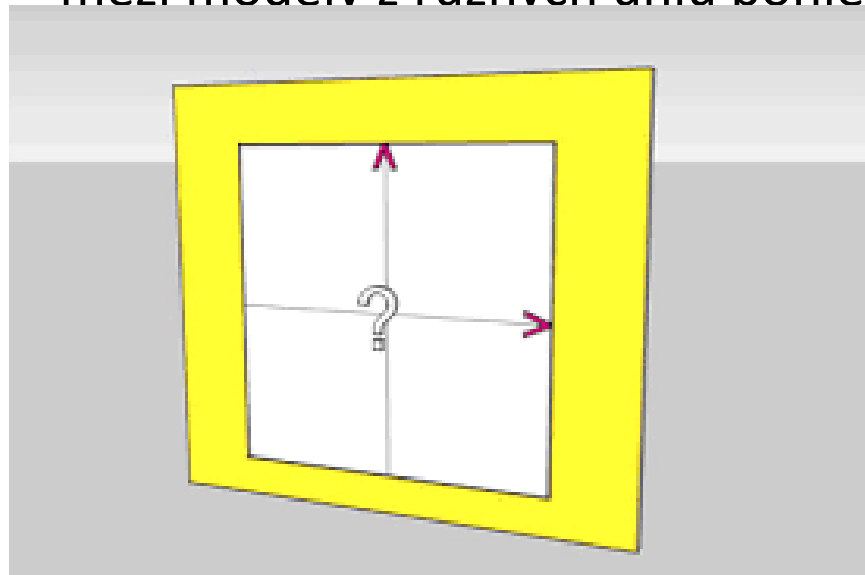


# Princip modelování

- Specifikovat systém nezávisle na platformě na které je systém vystavěn
- Specifikovat platformy
- Vybrat konkrétní platformu
- Transformovat systém podle zvolené platformy
- **MDA bez transformací nemá smysl !**

# Tranformace

- Př. : jazyk ATL = Alternativní transformační jazyk
- Typy transformací
- Vertikální – mezi modely různé abstrakce
- Horizontální – mezi modely z různých úhlů pohledu



# Tranformace

•Reálný svět  $\leftarrow \rightarrow$  Objektové paradigma  $\leftarrow \rightarrow$  Scala/Java/Smalltalk

•Konceptuální  
model

kód

model

Logický

•CIM, PIM

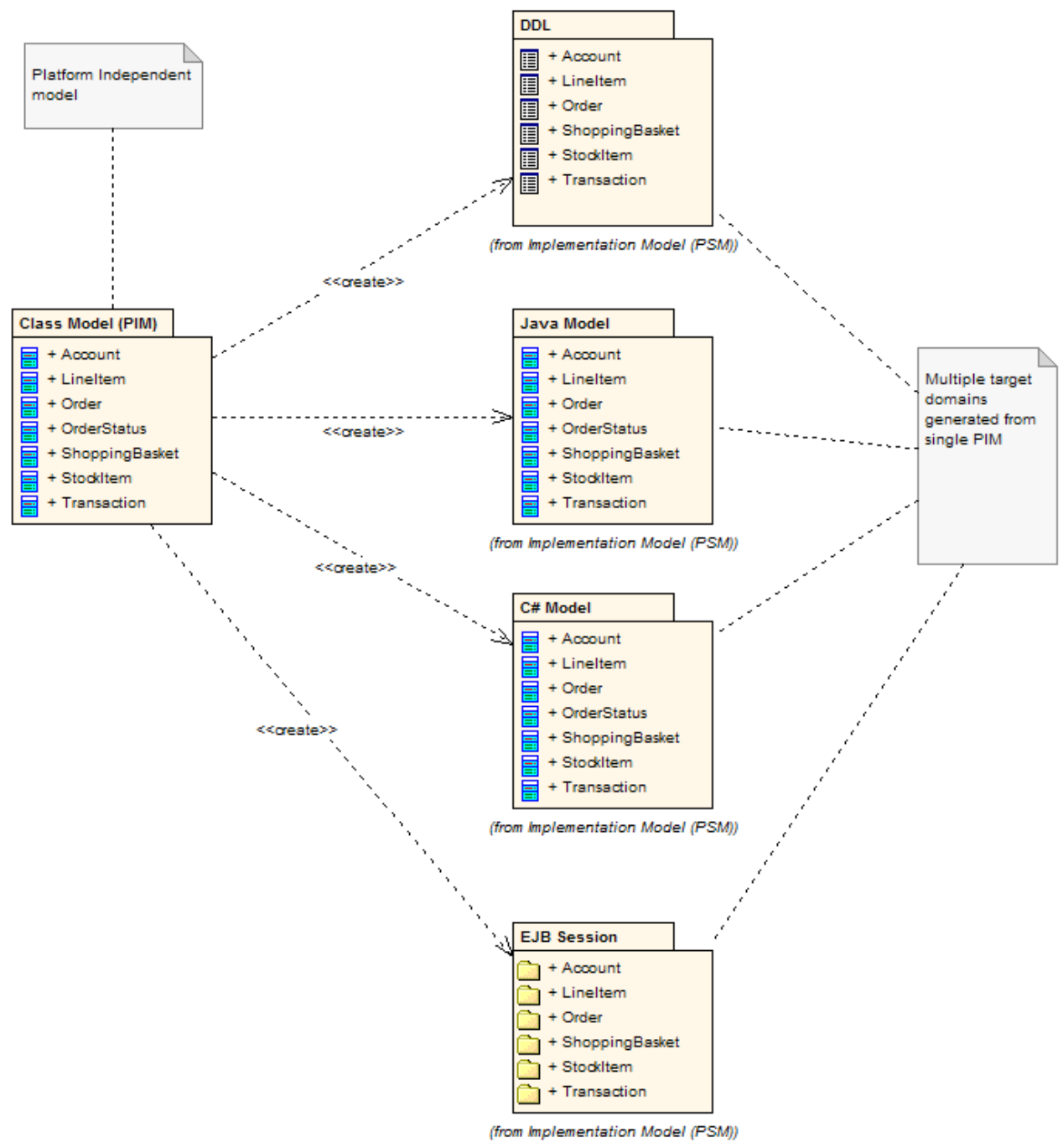
PSM, PDM

PIM, PSM



# Další příklady transformace

- produkty sparxsystems (enterprise architect) umí transformovat tyto věci :
  - DDL - Transforms platform-independent class elements to platform-specific table elements.
  - EJB Entity - Transforms platform-independent class elements to packages containing the class and interface elements that comprise an EJB Entity Bean.
  - EJB Session - Transforms platform-independent class elements to packages containing the class and interface elements that comprise an EJB Session Bean.
  - Java - Transforms platform-independent elements to Java language elements.
  - C# - Converts a PIM to a standard C# implementation set.
  - XSD - Transforms platform-independent elements to XSD elements.



**Děkuji za pozornost**

Martin Tomášek

