

# O čem bude řeč?

- Problémy vývoje software
- Model-Driven Architecture a její klíčové komponenty
  - Meta-Object Facility, XML Metadata Interchange, MOF Query/Views/Transformations
- Architektura MDA nástrojů

# Problémy vývoje software

- Produkty fáze analýzy projektu se stávají mrtvými dokumenty
- V “enterprise” aplikacích je nutné využití několika technologií (databáze, aplikační servery (J2EE, .NET), atd.)
  - Nutnost integrace, udržování integrity artefaktů
- S rostoucí flexibilitou technologií roste jejich složitost

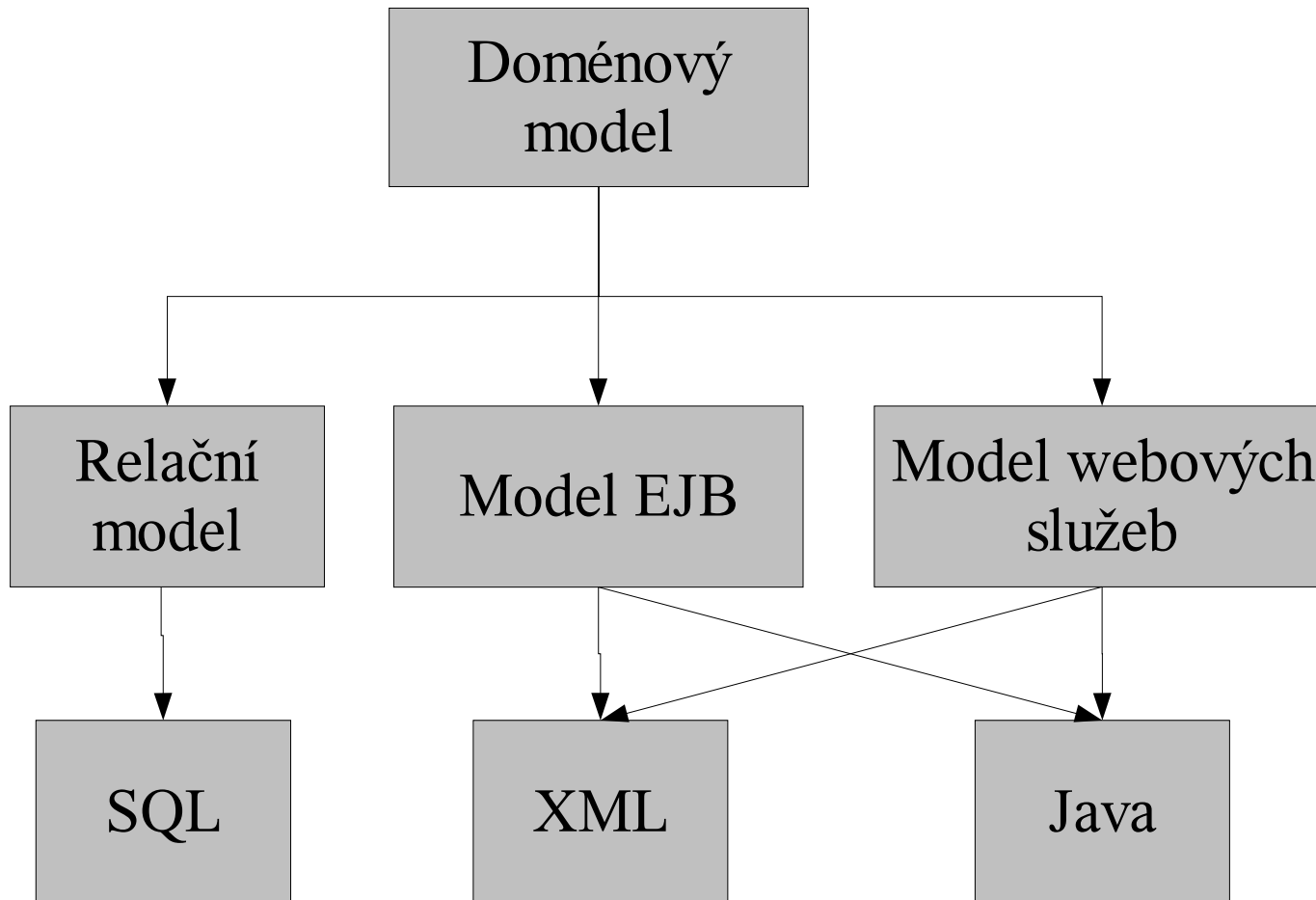
# Řešení

- Umožnit práci na vyšší úrovni abstrakce (jako ve fázi analýzy)
  - Postupovat od technologicky nezávislých modelů dolů přes modely specifické pro jednotlivé technologie až k samotnému zdrojovému kódu aplikace
  - Poskytnout možnost parametrizace jednotlivých transformací mezi modely a zdrojovým kódem
- O tom je Model-Driven Architecture

# Model-Driven Architecture (MDA)

- Main-stream vize OMG (Object Management Group)
- Framework standardů pro podporu modelově řízeného vývoje
- Centrální myšlenkou je umožnění tvorby modelů na různé úrovni abstrakce a definice jejich vzájemných transformací

# Model-Driven Architecture



# Základní komponenty MDA

- Modelovací jazyky pro tvorbu platformově nezávislých a platformově závislých modelů
- Jazyk pro formální popis transformací
- Jazyk pro formální popis modelovacích a textových jazyků => čtyři meta-úrovně

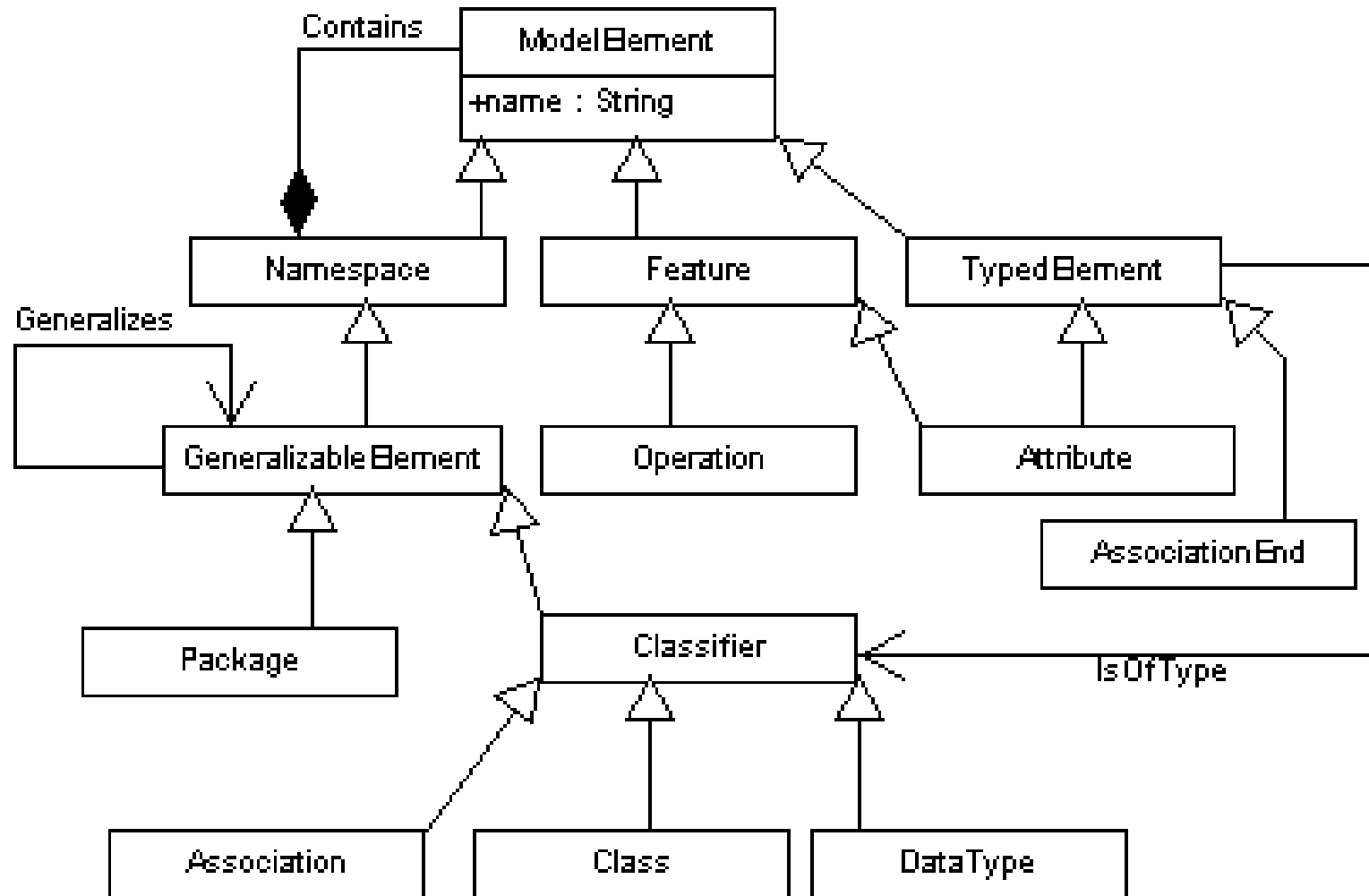
# Meta-úrovně

- Instance (M0) – data, konkrétní instance objektů  
(např. záznam v DB tabulce - Jan Novák)
- Modely (M1) – definice struktury instancí  
(např. definice tabulky Člověk={jméno,příjmení})
- Meta-modely (M2) – definice jazyků/konstrukcí  
pro tvorbu modelů/tříd  
(např. Tabulka={0 až n sloupců})
- Meta-meta-model (M3) – definice jazyka  
pro popis jazyků

# Meta-Object Facility (MOF)

- OMG standard pro meta-meta-model
- Velmi podobný části jazyka UML určené k modelování tříd
- Abstraktní jazyk (nedefinuje svou gramatiku ani grafickou notaci)
- Určen k popisu abstraktních syntaxí (nikoliv gramatik)

# Hlavní konstrukce MOFu



# Související standardy

- XML Metadata Interchange (XMI)
  - Definuje mapování MOFu do XML
  - Slouží pro výměnu meta-modelů a jejich instancí
- Human Usable Textual Notation (HUTN)
  - Podobné XMI, ale produkovaný dokument je člověkem lépe čitelný, než XML
- UML Profile for MOF
  - Mapování MOFu na grafickou notaci jazyka UML

# MOF Query/Views/Transformations (MOF QVT)

- Jazyk pro formální popis transformací
- Lze jej použít k popisu transformací:
  - Mezi modely na různé úrovni abstrakce
  - Normalizačních, či jiným způsobem refaktorujících stavající model
  - Mezi abstraktní a konkrétní syntaxí jednotlivých jazyků
- Několik návrhů na realizaci – rozdíly ve stupni deklarativnosti, symetrii, způsobu parametrizace a trasovatelnosti

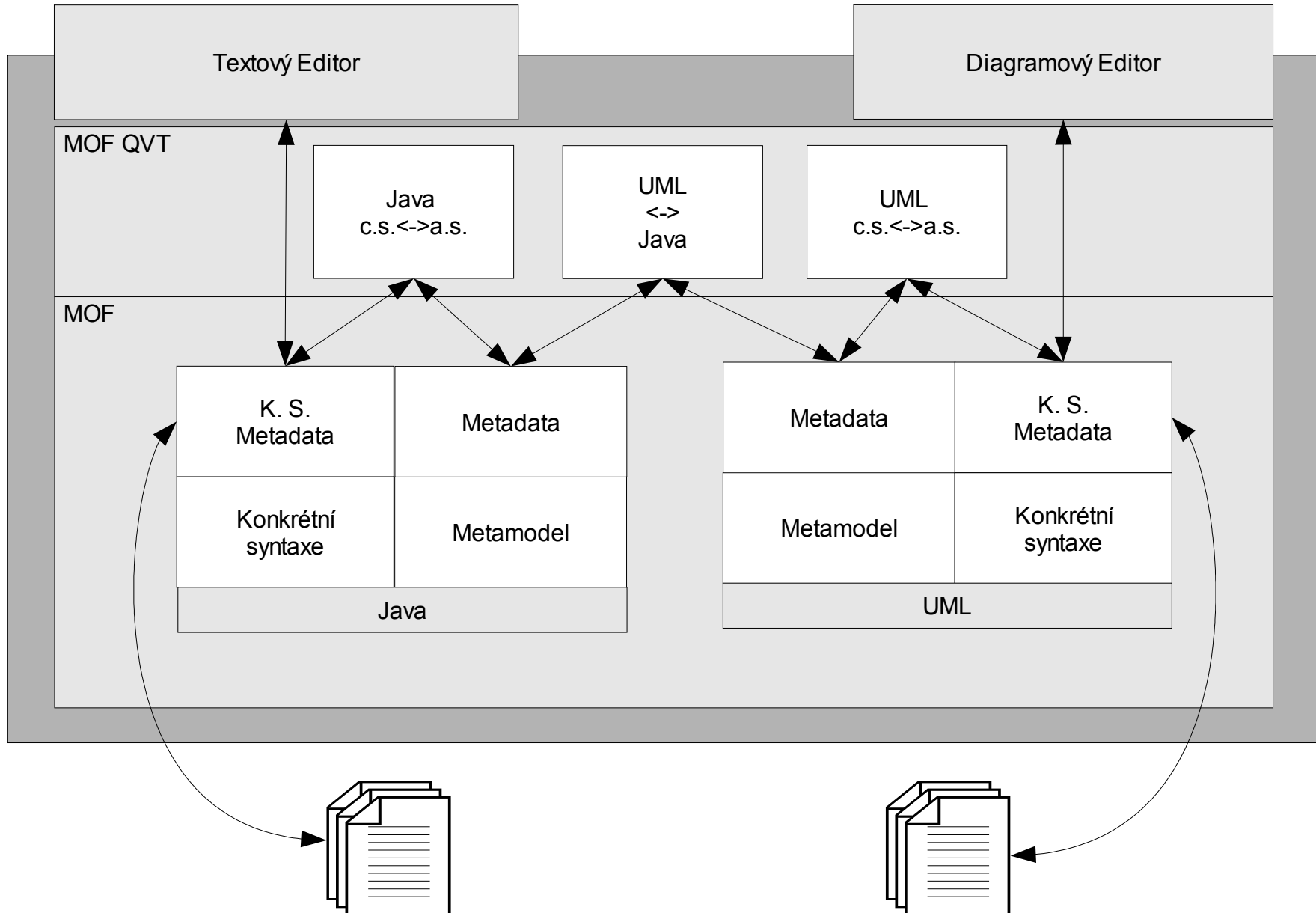
# Modelovací jazyky

- Unified Modeling Language (UML)
  - Pokrývá modelování struktury (diagramy tříd, instancí) a do jisté míry i modelování dynamiky (diagramy akcí, use-case diagramy)
  - Objektově orientovaný
- Nutnost neomezovat se pouze na UML

# Konkrétní syntaxe

- Textové (gramatiky) a grafické notace
- V současné době v OMG standardech pouze slovní popis
  - Nejednoznačné a nepřesné – volnost k různým interpretacím
  - Nelze jej zpracovat počítačem
- Nutnost formálního popisu (např. pomocí MOFu)

# Cílová architektura MDA nástrojů



# Závěr

- MDA – nové paradigma vývoje software
  - Umožní tvorbu do značné míry deklarativně rozšiřitelných vývojových nástrojů
  - Modely vznikající v úvodních fázích realizace projektu budou klíčovými, živými artefakty (možnost zpětné aktualizace)
  - Eliminuje nekreativní práci při vývoji aplikací