

---

# **WebML**

**(Web Modeling Language)**

**© Radek Ošlejšek**  
**Fakulta informatiky MU**  
oslejsek@fi.muni.cz

# WebML - účel

---

- WebML má za cíl poskytnout „strukturovaný“ přístup k návrhu webových aplikací, která intenzivně pracují s daty
- Ucelená sada modelů by měla pomoci při vytváření kvalitních webů
- WebML pokrývá všechny aspekty návrhu webových stránek
- Použití starých a nekoherentních metod pro návrh webů je špatný přístup
  
- viz [webml.org](http://webml.org)

# Jazyky a notace pro modelování webu

---

- WebML není jediný, existují např.:
  - HDM – W2000
  - RMM
  - OOHDM
  - ARANEUS
  - STRUDEL
  - TIRAMISU
  - Hera
  - UML Web Application Extension
  - UML-based Web Engineering (UWE)
  - ACE
  - WebArchitect
  - OO-H

# Cíle WebML

---

- Vytváření datově intenzivních webů
  - velký objem zpracovávaných dat
  - rozhraní určená pro širokou veřejnost
    - explanatory
    - browsing-oriented
    - personalized (1 to 1)
  - proměnlivý obsah, struktura, navigace, prezentace
- WebML není vhodné pro
  - malé weby (osobní stránky, ...)
  - statické weby
- WebML = modelovací jazyk + metodika + CASE nástroje

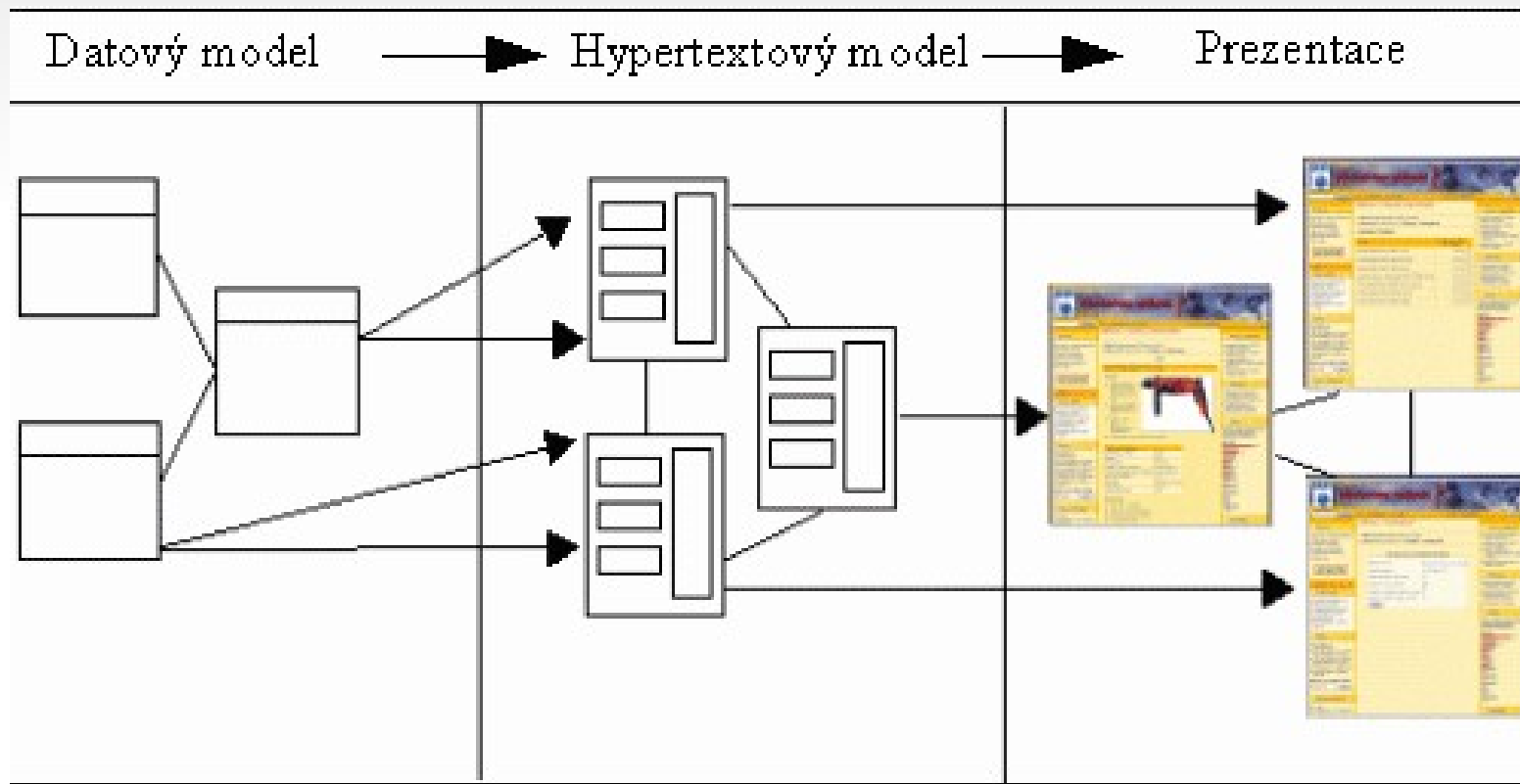
# Vlastnosti WebML

---

- Méně robustní než UML
- Zaměřeno čistě na webové aplikace
- Podpora v CASE nástrojích:
  - WebRatio – podporuje generování kódu z navržených modelů
  - MS Visio – obsahuje šablony diagramů

# Základní modely WebML

- Datový model
- Hypertextový model
- Prezentační model

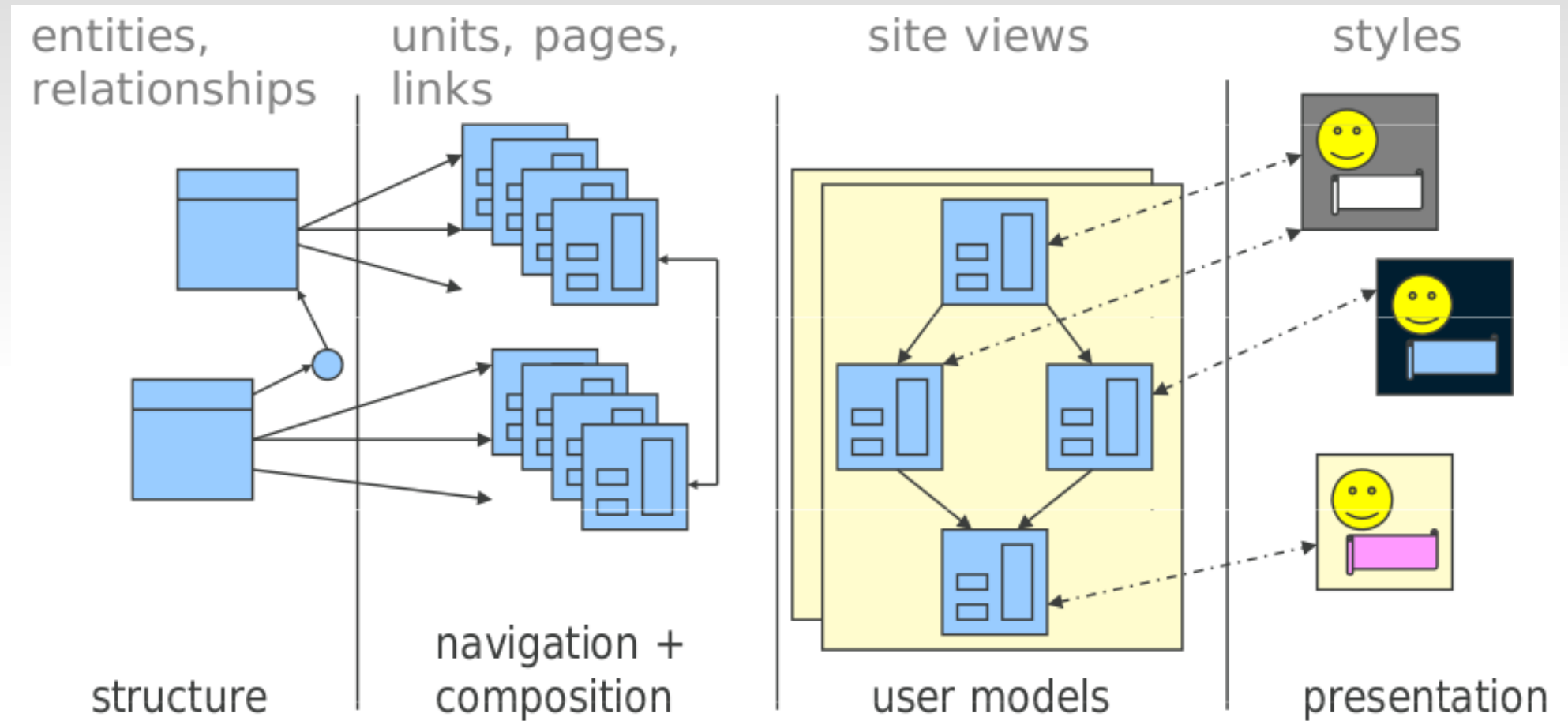


# Detailní modely WebML

---

- Datový model
  - Structure model – organizace dat
  - Derivation model – definice redundance na datech
- Hypertextový model
  - Composition model – definice stránek pomocí kolekce podstránek a elementárních publikačních jednotek
  - Navigation model – definice propojení mezi stránkami a jednotkami
- Uživatelský model
  - Siteviews – oprávnění, přístupová práva
- Presentation model
  - umístění jednotek na stránce a definice jejich vzhledu

# Koncepty WebML





---

# Datový model

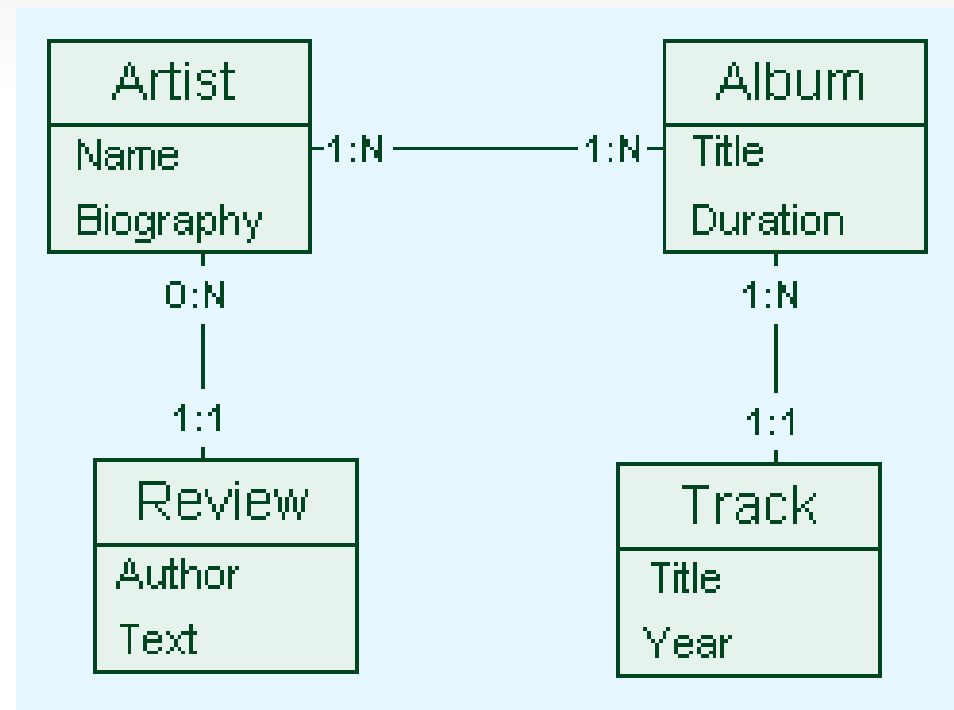
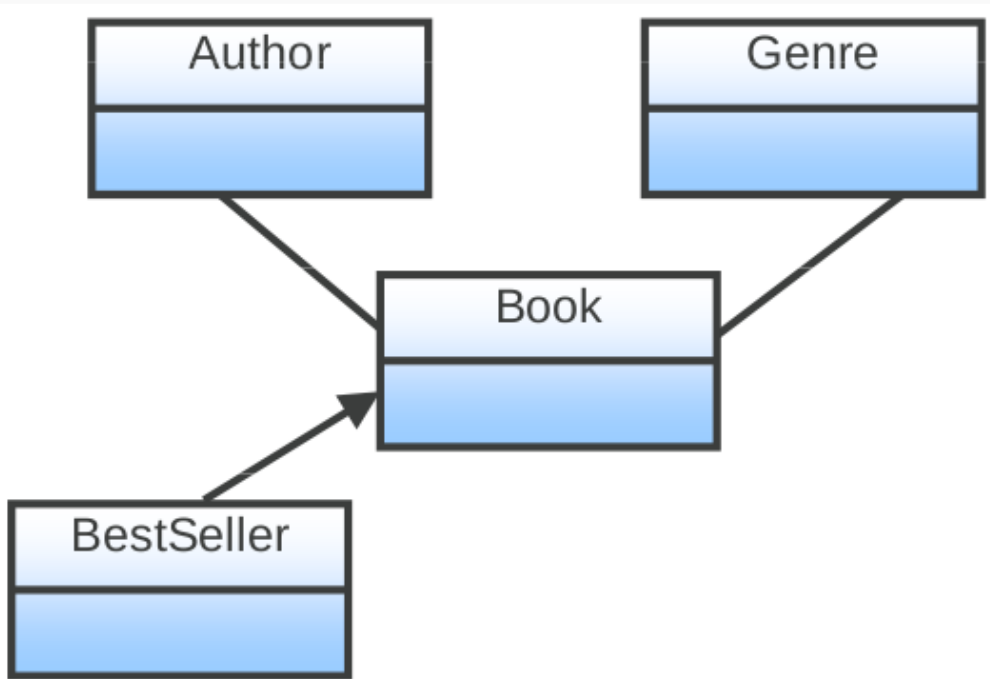
# Structure Model (i)

---

- Řeší otázku, jaké objekty (data) jsou prezentovány na webech a jak jsou provázány?
- Modeluje:
  - Entity – typ objektu v aplikační doméně
  - Atributy – skalární hodnoty (vlastnosti) entit
  - Vztahy (relace) – propojení entit
  - Hierarchie IS-A (dědičnost) – klasifikace objektů a jejich seskupování
- Kompatibilní s entitně-relačním diagramem (ERD) a UML diagramem tříd
- Konceptuální model, který se velice jednoduše převádí do konkrétní relační databáze (Oracle, MySQL, PostgreSQL, ...)

# Structure Model (ii)

- Zjednodušený entitně-relační model
  - pouze binární relace mezi entitami
  - dědičnost
  - jednoduché typované atributy



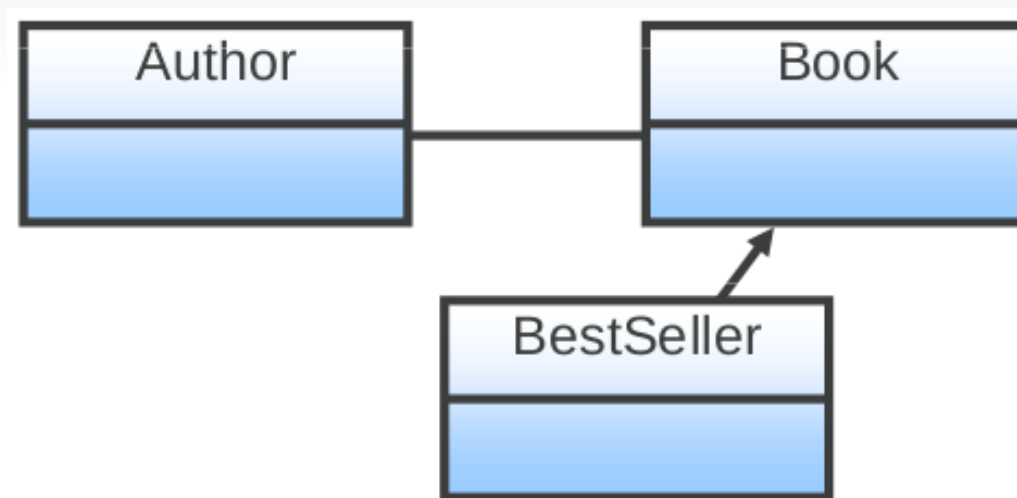
# Derivation Model

---

- Analogie s pohledy (views) v relačních databázích
- Přidává redundanci na datech řízeným způsobem.  
Redundance určuje přesnou podobu požadovaných informací ve webové aplikaci
- Redundantní data mohou být snadno popsán pomocí WebML-OQL (Object Query Language) – dost připomíná OCL (Object Constraint Language) z UML

# Derivation Model – příklad

- Příklad:
  - `BestSeller := Book where Book.Sales > 50.000`
  - `Author.BooksNumber = count(self.Author2Book)`



---

# Hypertextový model

# Hypertextový model (i)

---

- Modeluje chování celé webové aplikace
- Je považován za hlavní přínos WebML
  - výstižnost, jednoduchost, ...
- Dva sub-modely:
  - composition model
  - navigation model

# Hypertextový model (ii)

---

- Otázka 1: Jaké informace jsou publikovány v hypertextových uzlech?
- Odpověď 1: obsahové jednotky – content units (Composition Model)
  
- Otázka 2: Jak jsou hypertextové uzly propojeny?
- Odpověď 2: odkazy – links (Navigation Model)
  
- Otázka 3: Jak je hypertext rozdělen na stránky?
- Odpověď 3: stránky – pages (Composition Model)



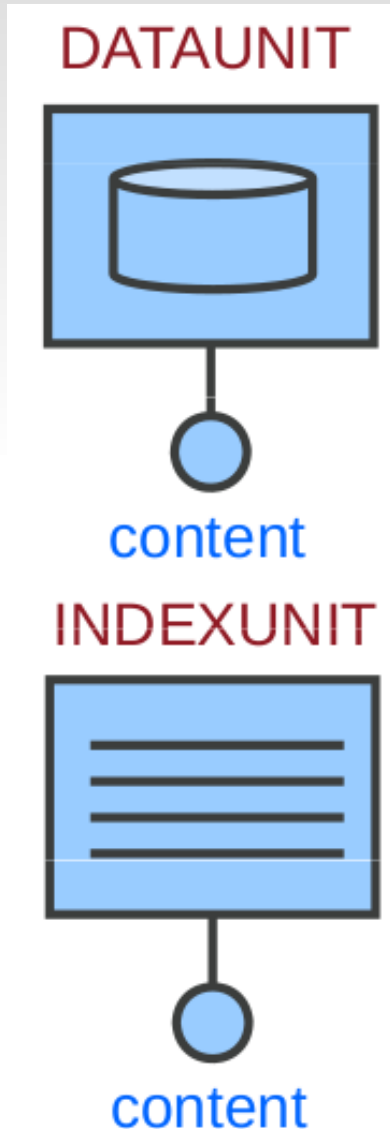
# Composition Model

---

- Definuje složení webové aplikace z jednotlivých stránek a složení stránek z předdefinovaných elementů (obsahových jednotek – content units).
- Pro každou obsahovou jednotku existuje grafické vyjádření.
- Propojením obsahové jednotky s entitou ze strukturního modelu se určí typ dat.
- Příklady obsahových jednotek:
  - DataUnit
  - MultiDataUnit
  - IndexUnit
  - MultiChoiceUnit
  - HierarchicalUnit
  - ScrollerUnit
  - EntryUnit

# Comp. Model – obsahové jednotky (i)

- Příklad obsahových jednotek



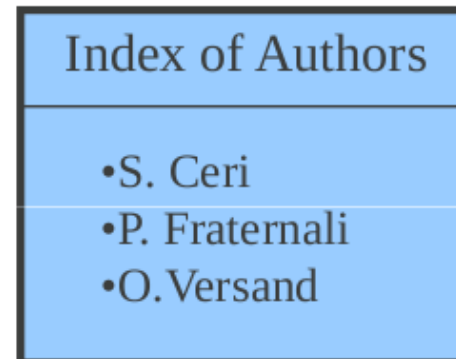
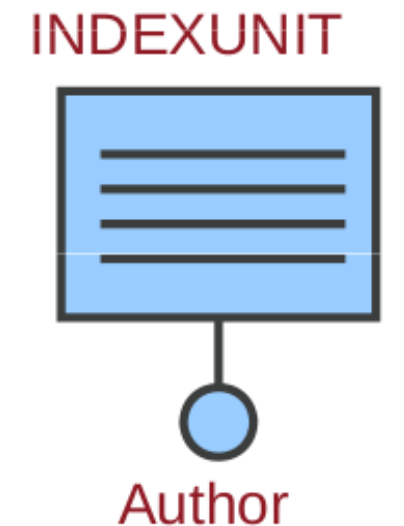
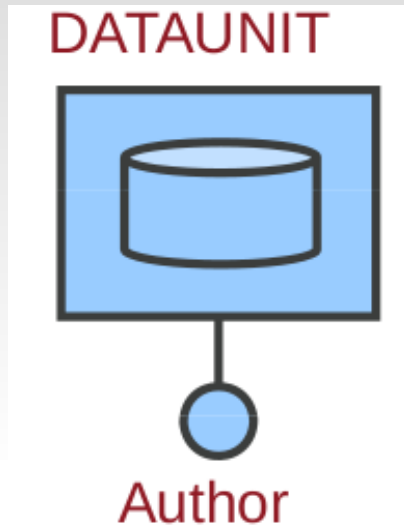
Určeno k publikování informací o JEDNOM objektu (např. AuthorDetail)



Určeno k publikování seznamu objektů (např. IndexOfAuthors)

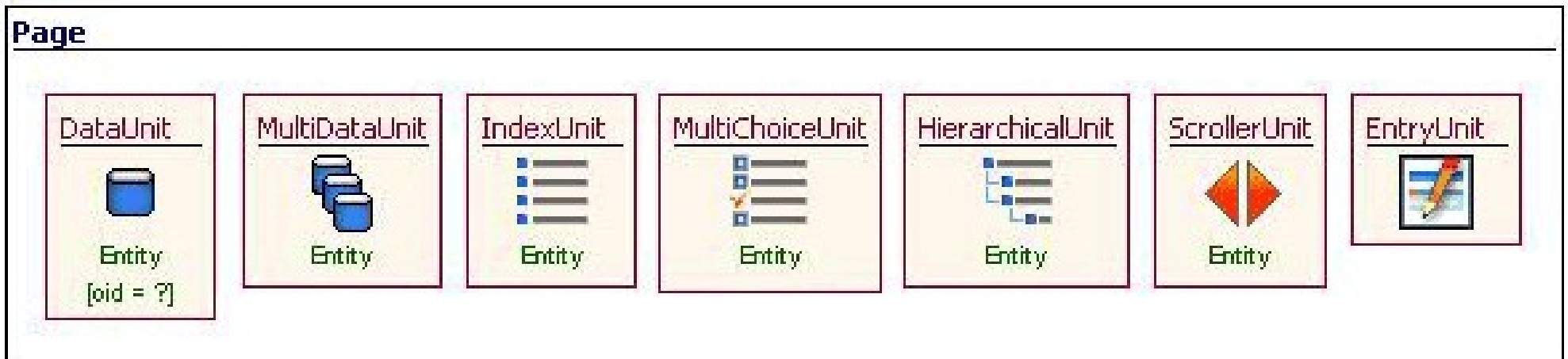
# Composition Model (ii)

- Příklad zobrazení obsahových jednotek



# Composition Model: stránky

- Stránka je strukturovaný kontejner obsahových jednotek a odkazů
  - možnost strukturovat na podstránky
  - abstrakce pro obrazovky, rámce, záložky, vrstvy, ...
  - umožňuje shlukovat relevantní informace za účelem efektivní komunikace
- Příklad:



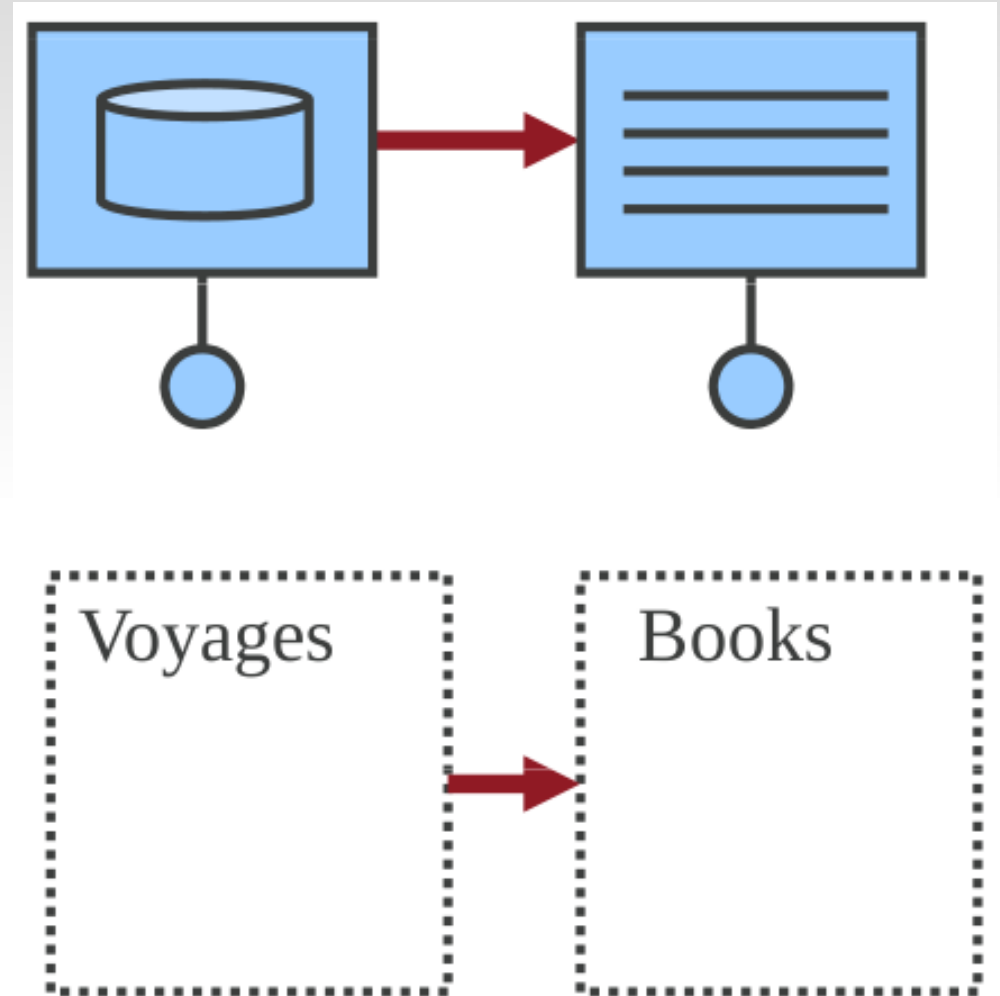
# Navigation Model: odkazy

- Sémantika odkazů:
  - přesun z jednoho místa na jiné
  - přenos informací z jednoho místa na jiné (navigační kontext)
  - spuštění výpočtu (side efect)

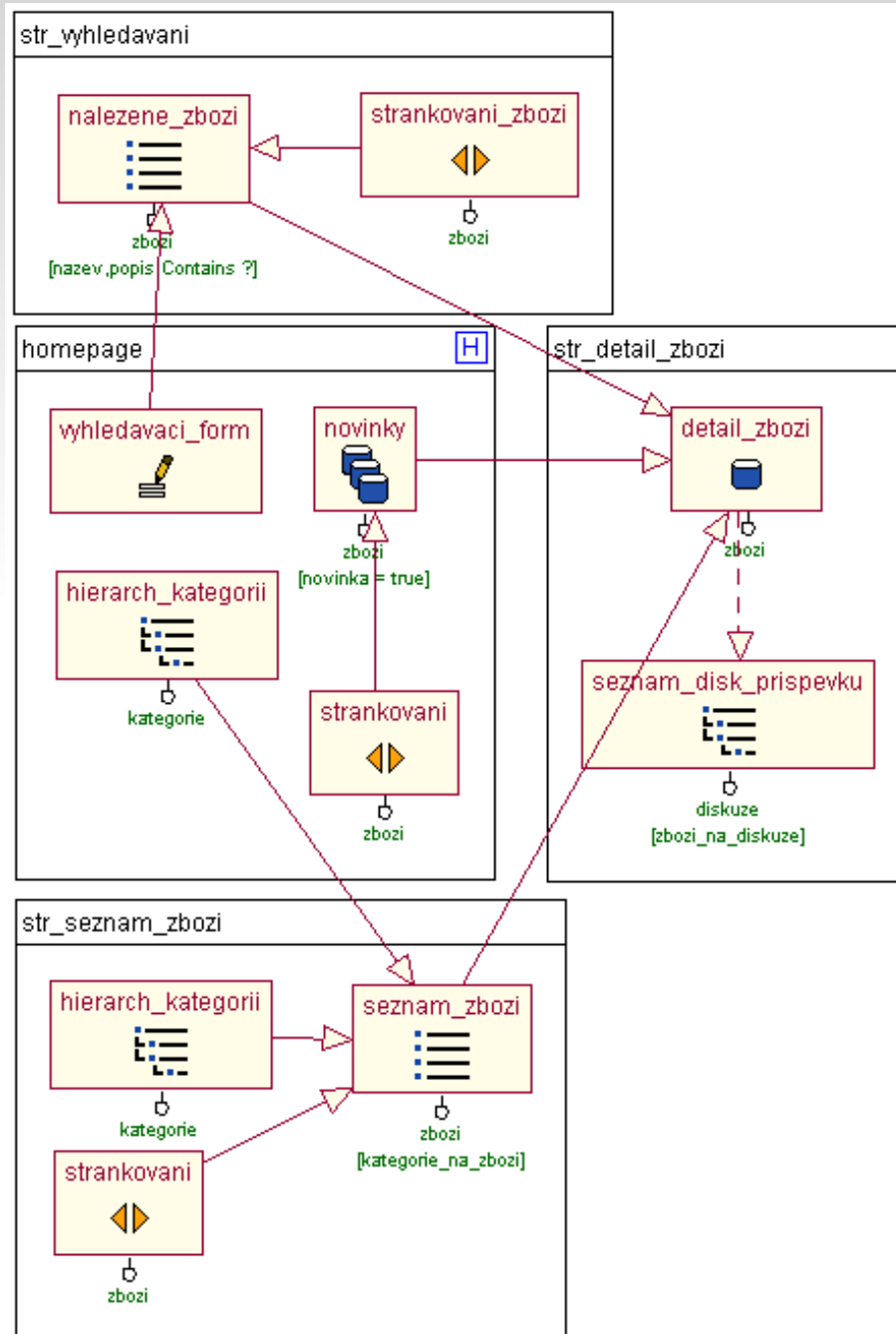


# Navigation Model: Typy odkazů

- Kontextové odkazy
  - mezi jednotkami
  - přenos kontextu
  
- Nekontextové odkazy
  - mezi stránkami
  - žádný kontext se nepřenáší

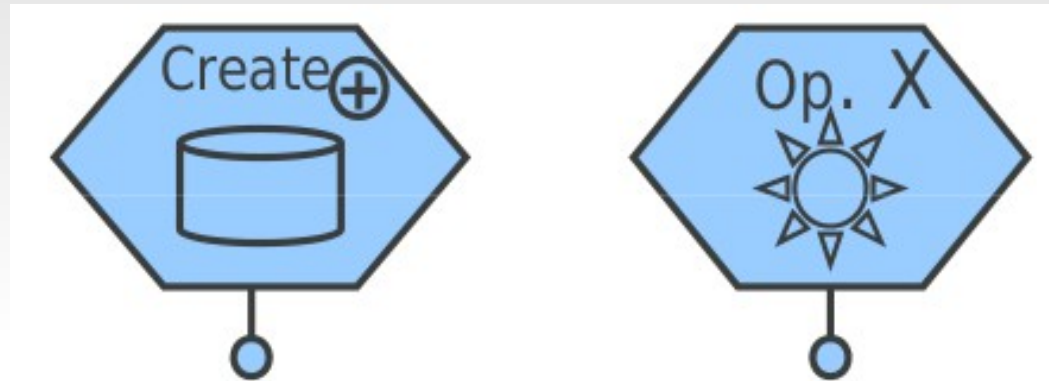


# Navigation Model: Příklad

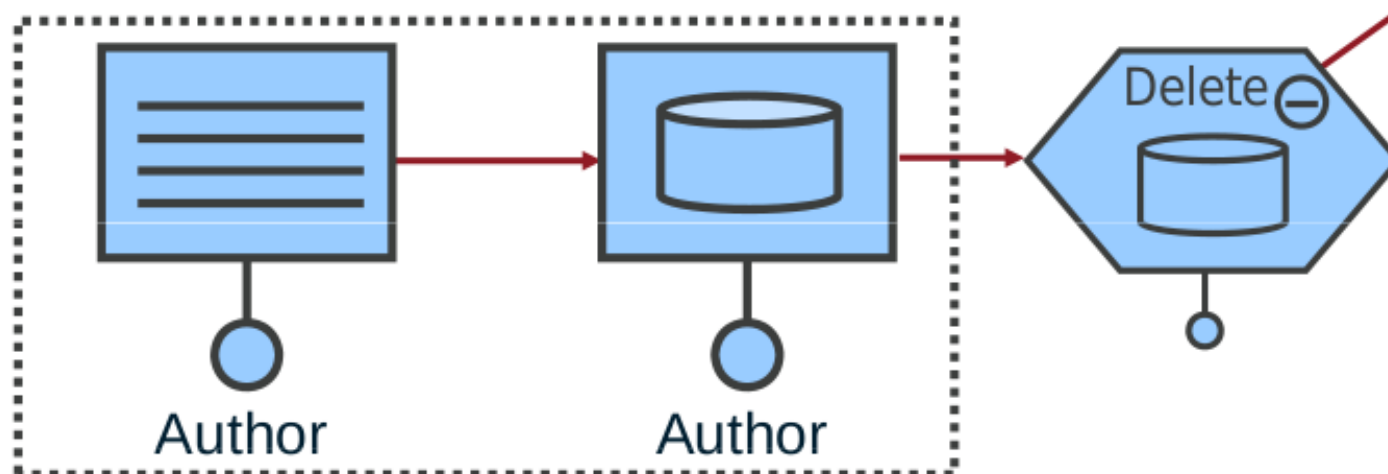


# Operace WebML

- WebML poskytuje několik předdefinovaných operací
- Mohou být dodefinovány další uživatelské operace



- Příklad: smazání autora





---

# Uživatelské modely

# Uživatelské modely – siteviews

---

- Uživatelské modely obsahují tzv. siteviews
- „Siteview“ je sada stránek, které konkrétní uživatel vnímá jako celý web (web site)
- Mohou být definovány různé „siteviews“ pro různé typy zařízení a skupiny uživatelů (uživatelské role)

---

# Prezentační model

# Prezentační model

---

- Vzhled a rozmístění prvků stránek bez ohledu na výstupní zařízení pomocí XML transformací
- Vstupem jsou datové a hypertextové model v podobě XML + CSS styly apod
- Výstupem jsou šablony stránek vygenerovaných pomocí XSLT

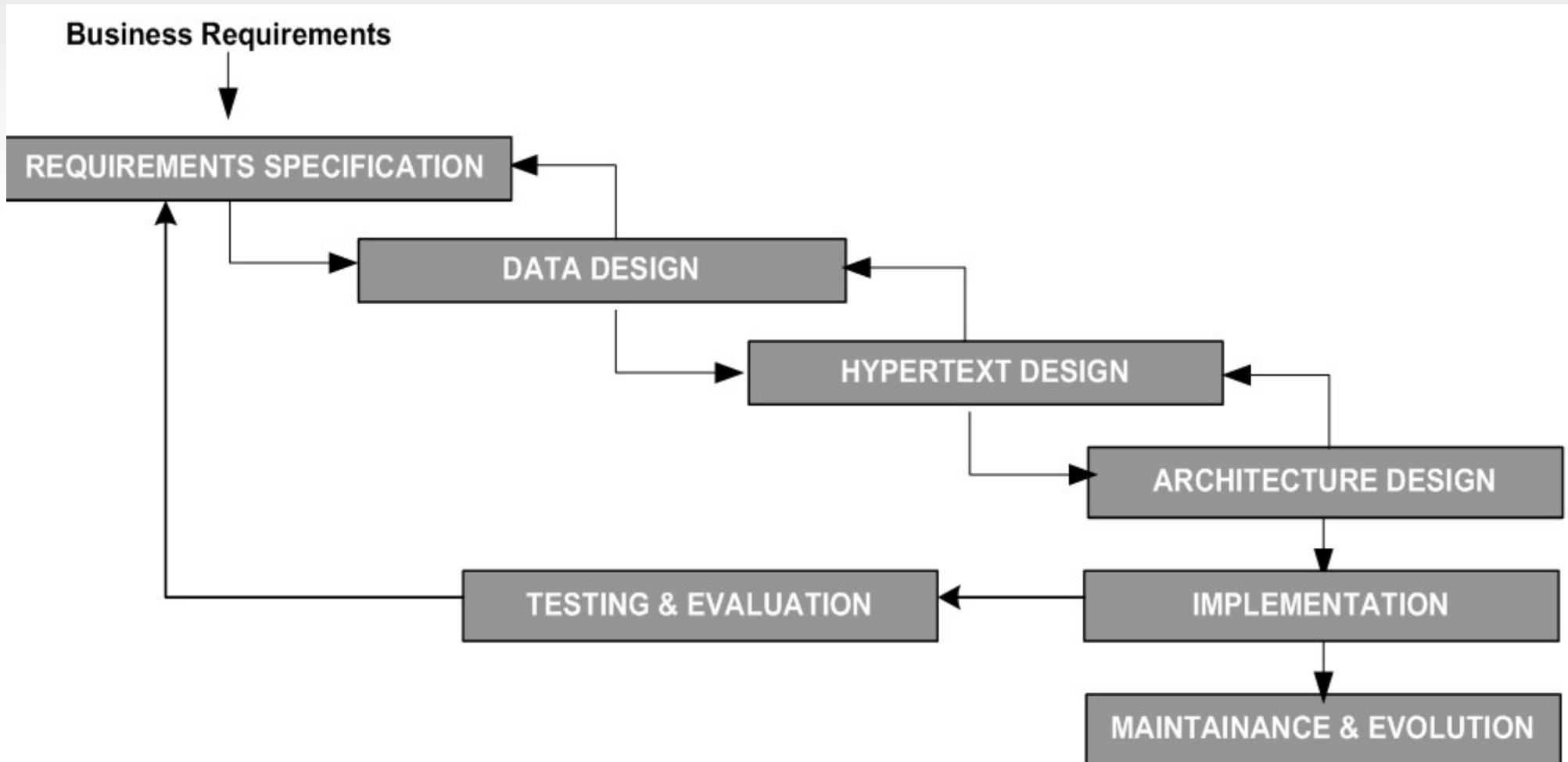
---

# WebML jako metodika

## WebML Development Process

# WebML Development Process

- Nedostatky klasických metodologií založených na UML:
  - nejsou zaměřeny na uživatelské rozhraní
  - nejsou zaměřeny na hypertextovou navigaci



# Shrnutí

---

- WebML je doménově specifický jazyk
  - není to UML ani MDA
  - ale má s nimi něco společného
- WebML se založeno na Model Driven Engineering
  - zaměřeno na datově intenzivní webové aplikace
  - automatické generování kódu webových aplikací
- Podpora v podobě CASE nástroje