

Porovnání relačních a OO datových modelů

Václav Papež

vpapez@kiv.zcu.cz

2012

Přehled prezentace

- Proč byla práce vytvořena
- Příklad v podobě mluveného slova
- Relační model
- Objektově model
- UML
- XML a XSD
- Souhrn transformačních pravidel
- Další sémantické rozšíření

Co je datový model

- Logicky strukturovaná množina dat
- Cílem modelu je vytvářet z dat interpretovatelné informace
- Různé způsoby popisu
 - Slovní vyjádření
 - Relační model
 - Objektový model
 - Ontologie

K čemu to, když je vše popsáno

- Málo komplexních “step by step” popisů
- Většina dokumentů se soustředí na jeden model a nehledá souvislosti s okolím
- Většinou je popsáno co lze, ne co nelze
- Málo dokumentů jde od definice po příklad
- Důležitý souhrn pro další popis transformací (OO model – RDF / OWL)
- Upřesnění častých omylů
 - ER model nemá grafickou syntaxi
 - Relace vs. vztah
 - UML je více než Class diagram
 - Class diagram má více vztahů než jen asociace

Příklad

- V jednom státě existuje několik univerzit
- Každá univerzita má několik fakult
- Každá fakulta má n studentů
- Jeden student může studovat více jak jednu fakultu (na různých univerzitách)
- Student má jméno a unikátní ID v rámci univerzity
- Fakulta/univerzita má jméno, studenty a děkana/rektora
- Děkan/Rektor má jméno, příjmení a jedinečné ID v rámci univerzity

Relační... 1/4

- Relační model
- x
- Entity-Relational model
- x
- Relační databáze
- x
- Relační algebra
- x
- Entity-Relational diagram
- x
- Databázový systém (SŘBD)

Relační... 2/4 (E. F. Codd)

- Doména

- *A domain is a set of values of similar type. A domain is simple if all of its values are atomic.*

- Relace

- *Let D_1, D_2, \dots, D_n be n ($n > 0$) domains (not necessarily distinct). The Cartesian product $\times\{D_i; i=1, 2, \dots, n\}$ is the set of all n -tuples $\langle t_1, t_2, \dots, t_n \rangle$ such that $t_i \in D_i$ for all i . A relation R is defined on these n domains if it is a subset of this Cartesian product. Such a relation is said to be of degree n .*
- Relace není vztah! Relation x Relationship

- Atribut

- *For each tuple component we associate not only its domain, but also its distinct index. This we call an attribute.*
- *The n distinct attributes of a relation of degree n distinguish the n different uses of the domains upon which that relation is defined. A tuple then becomes a set of pairs $(A:v)$, where A is an attribute and v is a value drawn from the domain of A , instead of a sequence (v_1, v_2, \dots, v_n) .*
- *A relation then consists of a set of tuples, each tuple having the same set of attributes.*

- Schéma

- *A schema for relation R with degree n is a list of unique attribute names.*

Relační... 3/4

- Primární (kandidátní) klíč
 - *K is a candidate key of relation R if it is a collection of attributes of R with the following time-independent properties.*
 - (1) *No two rows of R have the same K-component.*
 - (2) *If any attribute is dropped from K, the uniqueness property (1) is lost.*
- Relační model
 - Kolekce relací v tabulární podobě
 - Insert – Update – Delete
 - Relační algebra
- Relační databáze
 - *A relational database is collection of data represented by collection of time-varying tabular relations.*

Relační... 4/4

- Relační algebra
 - Soubor operací
 - UNION
 - INTERSECTION
 - SET DIFFERENCE
 - THETA-SELECT
 - PROJECTION
 - THETA-JOIN
 - DIVIDE
- Relační model vs. Entity-Relational Model
 - Relace vs. Vztahy mezi entitami
 - Struktura hodnot vs. Struktura entit
 - Kartézský produkt domén (relace) vs. Kartézský produkt entit (vztah)
 - ER model má explicitně vyjádřené kardinality a provázání entit

Konstrukce ER modelu 1/3

- Jednoduchý význam či kontext pro člověka nemusí být triviální pro strojové zpracování
- Základní pravidla
 - Podmět – Entita
 - Přísudek – Relace
 - Předmět – Atribut
- Příklad
 - University (name, rector, faculties)
 - Faculty (name, dean, students)
 - Dean (id, name, surname)
 - Rector (id, name, surname)
 - Student (id, name, surname)

Konstrukce ER modelu 2/3

- Po odstranění „neatomičnosti“ atributů, využití operací NATURAL-JOIN a vyčlenění primárních klíčů
 - University (**name**, **rector_id**, faculties)
 - Faculty (**name**, **dean_id**, students)
 - Dean (**dean_id**, name, surname)
 - Rector (**rector_id**, name, surname)
 - Student (**student_id**, name, surname)
- Vazby ve složení podmět–přísudek–předmět (subject-predicate-object)
 - ONE University has MANY Faculties
 - ONE University has ONE rector
 - ONE University has MANY students
 - ONE Faculty belongs to ONE University
 - ONE Faculty has ONE dean
 - ONE Faculty has MANY students
 - ONE dean direct ONE Faculty
 - ONE rector direct ONE University
 - ONE student study MANY Faculties
 - ONE student study MANY Universities

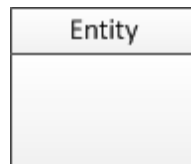
Konstrukce ER modelu 3/3

- Po vynechání redundantních vztahů
 - University 1!.....n! Faculty
 - University 1!.....1! Rector
 - Faculty 1!.....1! Dean
 - Faculty m.....n Student
- Tímto lze považovat ER model za hotový, grafická reprezentace do modelu nepatří
- Zpětným přepisem do vět lze „otestovat“ vyjadřovací schopnost modelu

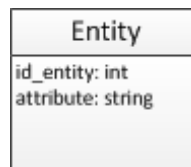
Grafická reprezentace 1/2

- Dle zvyklostí, atypicky, UML
- Dle zvyklostí:

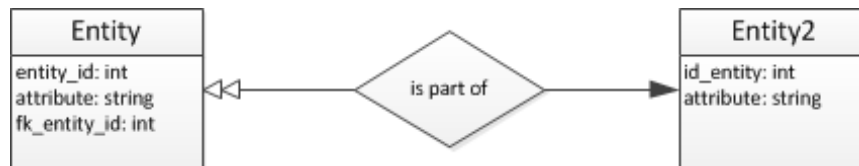
- Entita



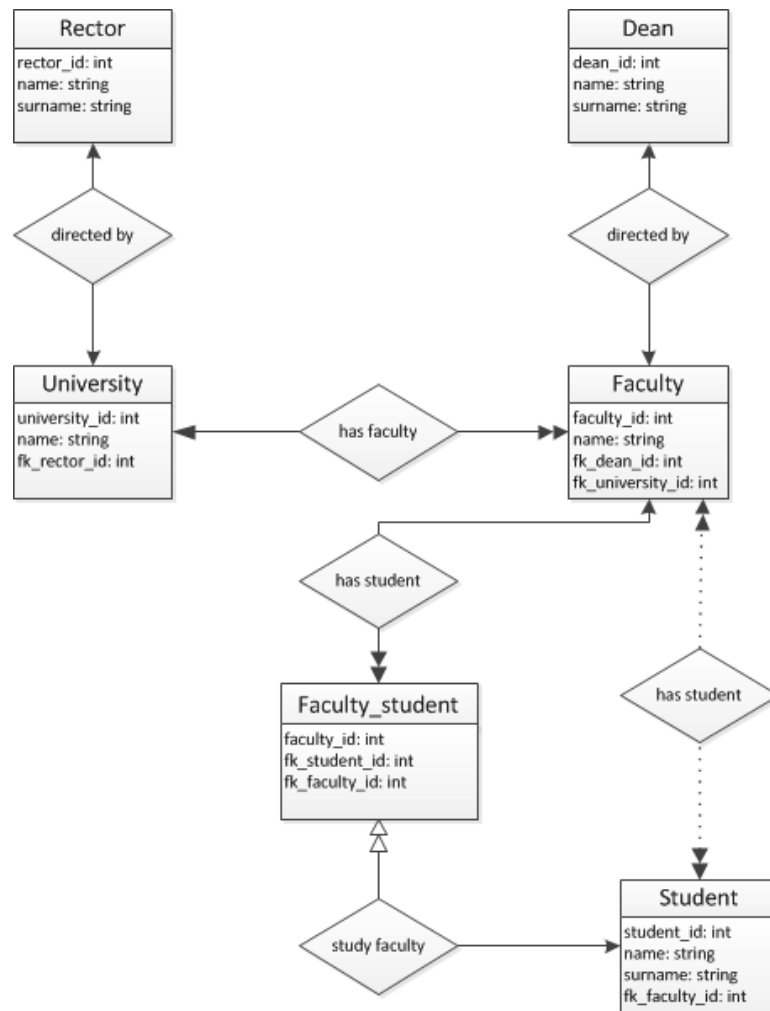
- Atribut



- Vazba a kardinalita



Grafická reprezentace 2/2



Zobecnění modelu

- Rektor, děkan i student mají stejné atributy – jsou konkretizací osoby
- Fakulta i univerzita jsou instituce
 - Institution (**institution_id**, name, director_id)
 - Person (**person_id**, name, surname)
 - Institution m!.....n! Person
- Nastávají problémy
 - Jak zaručit, že instituce bude mít jen jednoho ředitele?
 - Jak zaručit, že instituce bude mít alespoň jednoho studenta?

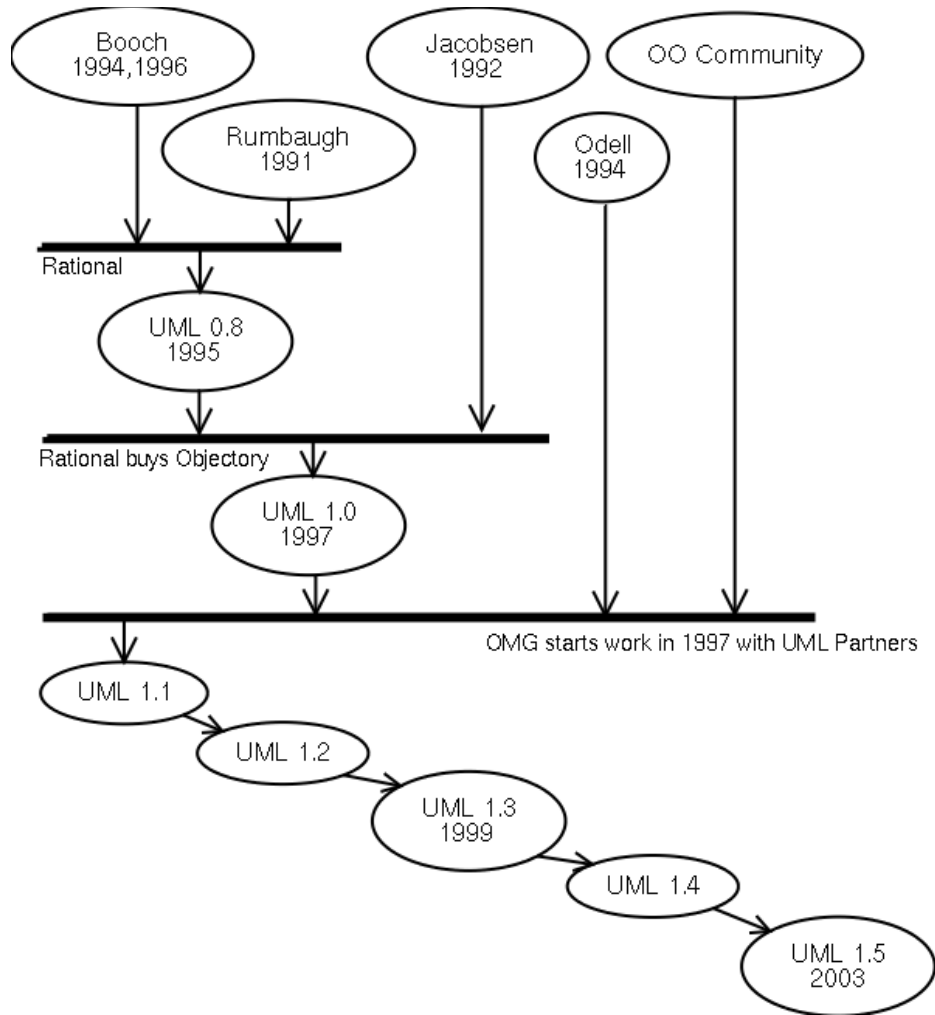
OO model 1/2

- The Core (Won Kim)
 - Objekt
 - Abstraktní reprezentace objektu reálného světa
 - OID – Logické x fyzické
 - Komplexní objekt (Set, List, Tuple konstruktory)
 - Atribute
 - Definuje stav objektu
 - Primitivní hodnota či objekt
 - Metoda
 - Definuje chování objektu a mění jeho stav
 - Třída
 - Šablona pro objekty
 - Dědičnost
 - Jednoduchá (strom) x vícenásobná (orientovaný graf)
 - Specializace x generalizace

OO model 2/2

- Třídy v *University* příkladu s využitím generalizace
 - Class Institute (name, director)
 - Class Person (name, surname, ID)
 - Class Rector extends Person ()
 - Class Dean extends Person ()
 - Class Student extends Person ()
 - Class University extends Institute (List of Faculties)
 - Class Faculty extends Institute (List of Students)
- Jak popsat vztahy mezi třídami?
- OO model má na rozdíl od ER modelu definovaný modelovací jazyk

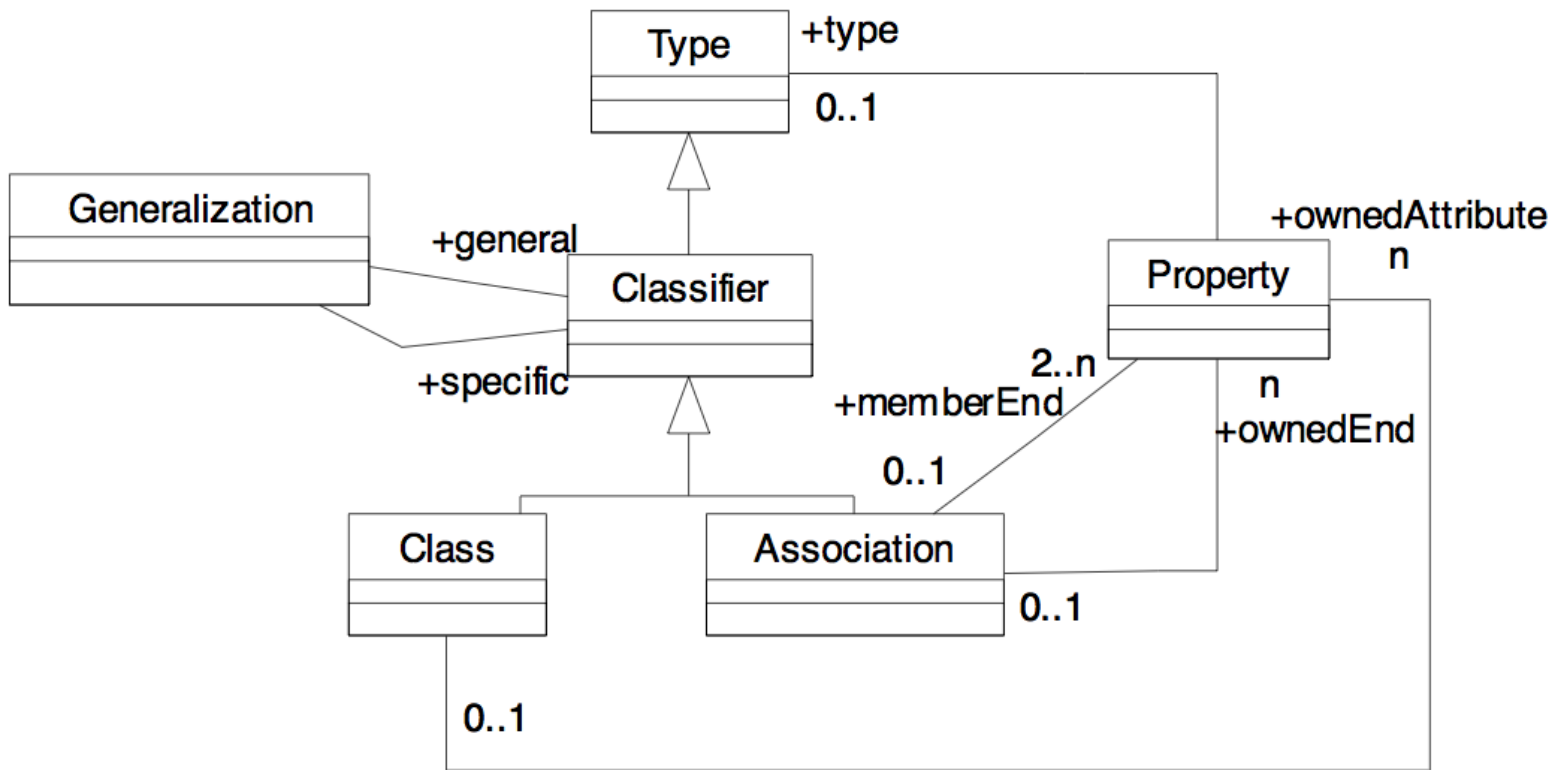
UML - Historie



UML 2.0 1/2

- 14 základních diagramů
 - Diagramy zaměřené na strukturu
 - Diagram tříd, komponent, balíčků, objektů, nasazení, vnitřní struktury, interakcí
 - Diagramy zaměřené na chování
 - Diagram aktivit, stavový, sekvenční, případů užití, časování, komunikace
- XML Metadata Interchange – jazyk pro popis UML založený na XML
- Problémy:
 - UML je nezávislé na implementaci, přesto designer musí brát v potaz implementační technologie (například násobná dědičnost)

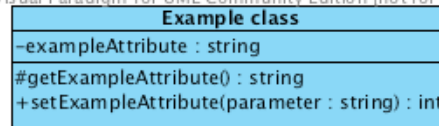
UML 2.0 2/2



Konstrukce diagramu tříd 1/2

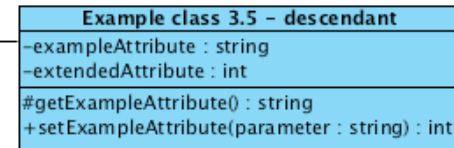
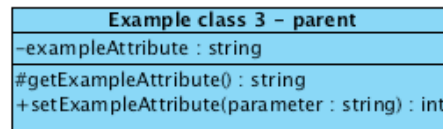
- Třída, atribut, metoda

Visual Paradigm for UML Community Edition [not for commercial use]



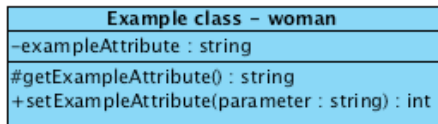
- Generalizace

Visual Paradigm for UML Community Edition [not for commercial use]

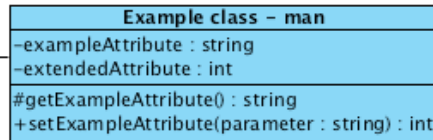


- Asociace, agregace, kompozice, multiplicita

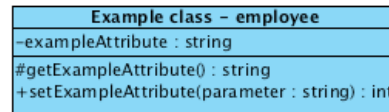
Visual Paradigm for UML Community Edition [not for commercial use]



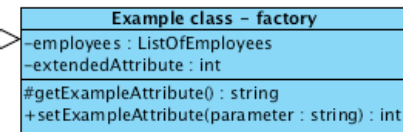
married



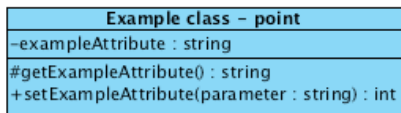
Visual Paradigm for UML Community Edition [not for commercial use]



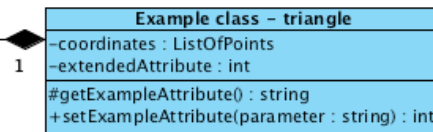
employed



Visual Paradigm for UML Community Edition [not for commercial use]



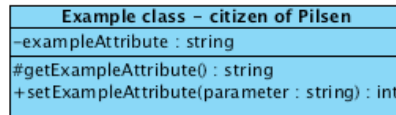
has



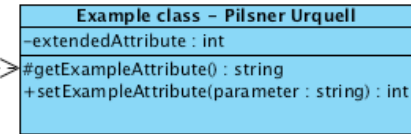
Konstrukce diagramu tříd 2/2

- Závislost

Visual Paradigm for UML Community Edition [not for commercial use]

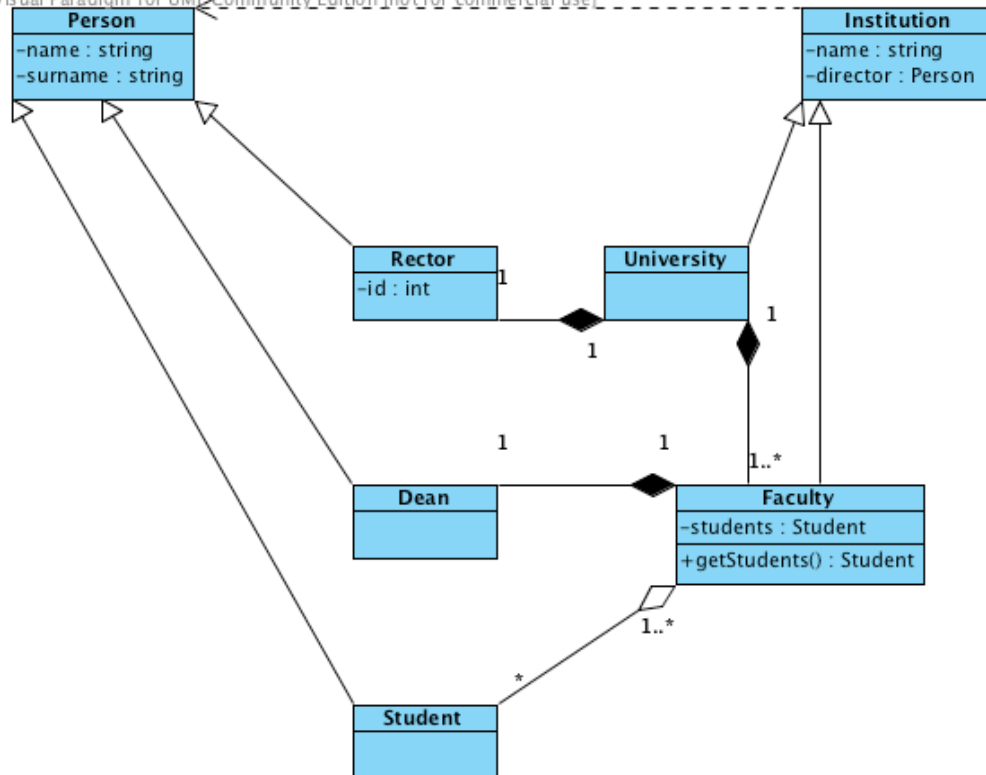


is proud on



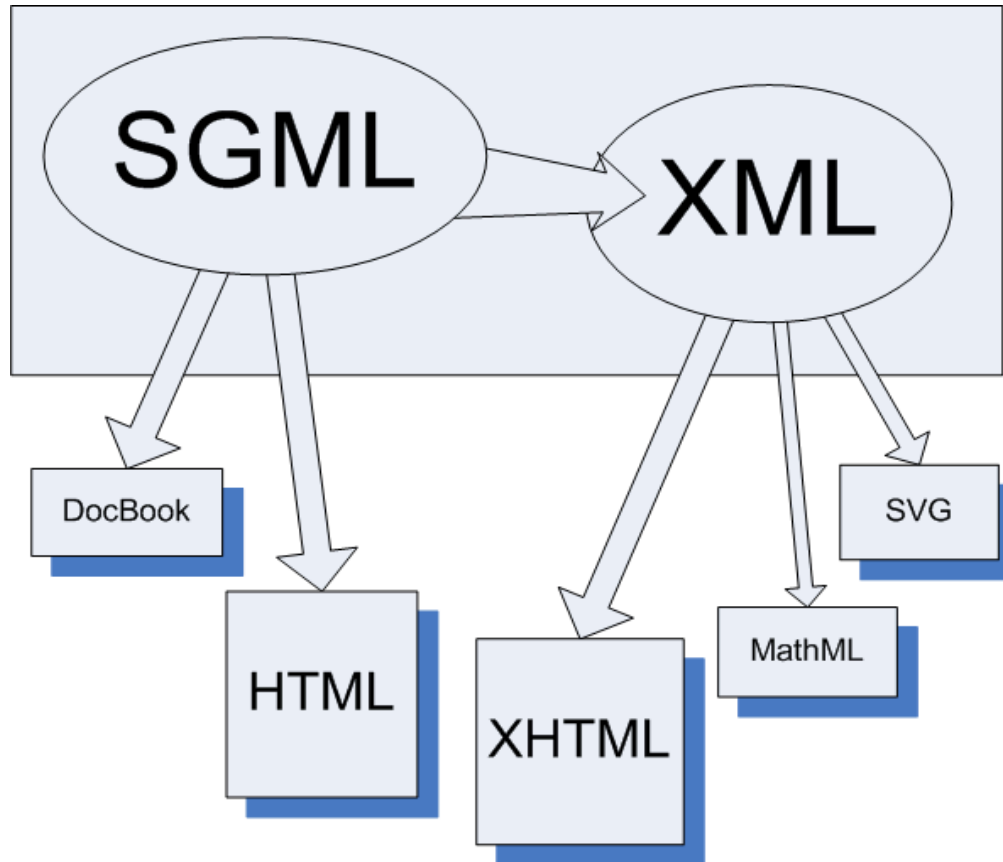
- Příklad

Visual Paradigm for UML Community Edition [not for commercial use]



XML 1/2

- Univerzální jazyk pro výměnu / přenos dat



XML 2/2

- XML umožňuje popis strukturovaných dat – proto je možné popsat i tabulkové struktury, tedy ER model
- XSD a DTD pro definici struktury XML dokumentů
- XSD umožňuje definovat dědičnost, zapouzdření, substituci apod. – proto je možné jím popsat OO model
- XML popisuje instance x XSD popisuje třídy
- UML diagram je popsateľný XML a UML je samo popsateľné UML modelem -> Obecně lze jazyk UML popsat v XML

XSD

- Entita – element
- Atribut – atribut nebo element
- XSD popisuje hierarchii elementů – complexType, simpleType
- Vztahy – dáno zanořením elementu ve struktuře
- Kardinalita – minOccurs, maxOccurs

XSD - Příklad

- `<xs:complexType name="students">`
- `<xs:sequence>`
- `<xs:element name="student" type="student"`
- `minOccurs="0" maxOccurs="unbounded"/>`
- `</xs:sequence>`
- `</xs:complexType>`
- `<xs:complexType name="student">`
- `<xs:sequence>`
- `<xs:element name="name" type="name"/>`
- `<xs:element name="surname" type="surname"/>`
- `</xs:sequence>`
- `<xs:attribute name="id" type="id"/>`
- `</xs:complexType>`
- `<xs:simpleType name="name">`
- `<xs:restriction base="xs:string">`
- `</xs:restriction>`
- `</xs:simpleType>`

Mluvené slovo – ER model

Mluvené slovo	ER model
Podmět	Entita / atribut
Podstatné jméno	Atribut
Sloveso	Vztah
Číslovka	Kardianlita
Ostatní	Silnější omezení, závislosti atd. Musí být vyvozeny z kontextu

Opačná transformace se řídí stejnými pravidly a zdravým rozumem.

ER model – UML Class diagram

ER model	UML Class diagram
Entita	Třída
Atribut	Atribut
Vztah	Asociace
Kardinalita	Multiplicita

UML Class diagram – ER model

UML class diagram	ER diagram
Třída	Entita
Atribut	Atribut
Zapouzdření	N/A
Metody	N/A (možnost nahradit databázovými metodami a triggerly)
Multiplicita	Kardianlita; multiplicita musí být zjednodušena ("?" zastupuje libovolné přirozené číslo kromě 1 či 0) <ul style="list-style-type: none">• 1 -> 1• ? -> n (omezení NOT NULL)• * -> n• 0..? -> n• ?..? -> n
Asociace	Vztah
Agregace	Agregace je transformována jako asociace se ztrátou informace "je součástí"
Compozice	Jako agregace; entita kompozitní třídy musí obsahovat cizí klíč
Závislost	N/A
Specializace/ Generalizace	N/A

UML Class diagram - XSD

UML Class diagram	XSD
Třída	Complex type
Atribut	Element / sequence type / complex content / complex type
Jednoduchý atribut	Element / attribute / simple content / simple type
Multiplicita	Minimal occurrence / maximal occurrence
Generalizace / Specifikace	Extension
Asociace	Group
Agregace/ Kompozice	Extension / Group / Group attribute

XSD – UML Class Diagram 1/2

XSD	UML Class Diagram
Complex type	Třída / Atribut
Sequence	N/A
All	Multiplicita 1
Choice	Multiplicita 0..*
Occurrence	Multiplicita ?..?
Element	Atribut / Třída
Attribute	Atribut
Simple content	Atribut
Complex content	Atribut
Extension	Generalizace / Specializace

XSD – UML Class Diagram 2/2

XSD	UML Class Diagram
Group	Agregace / Kompozice / Asociace / Generalizace
Attribute group	Agregace / Kompozice
Element any	N/A
Element anyAttribute	N/A
Substitution	N/A (alternativa – závislost / asociace / generalizace)

Pokračování

- Vytvoření základních transformačních pravidel pro prostředky sémantického webu, zejména pak pro konstrukce OWL 2.0

Děkuji za pozornost

- Dotazy?