Značkovací jazyky

Jan Růžička

ubuntu

Historie

- SGML
- HTML
- XML
- VRML
- TIFF
- ...

Současnost

- XML
- Schémata -> Mnoho nových jazyků, formátů

Budoucnost

- Kdo ví

ubuntu

Značky

- Různý charakter
- Ohraničující
- Uvozující
- Speciální znaky
- Escape sekvence

ubuntu

Výhody

- Čitelné člověkem i strojem (až na výjimky např. TIFF)
- Snadná modifikace souborů
- Stejný jazyk pro popis dat i algoritmů (chování)

Jmenné prostory

- Identifikují příslušnost značky nebo atributu
- xmlns:xsd="http://www.w3.org/2001/XMLSchema"
- xsd:double

Jmenné prostory / 2

- Možnost kombinovat více značek stejných názvů v jednom dokumentu

Značkovací jazyky

XML

ubuntu

XML

- eXtensible Markup Language
- Definice struktury dokumentu
- Univerzální jazyk
- Základ pro vývoj dalších jazyků

XML – Značky / 1

- Prvky, Elementy, Tagy

<NazevZnacky>

Obsah značky

</NazevZnacky>

XML – Značky / 2

<NazevZnacky/>

ubuntu

XML – Značky / 3

<NazevZnacky NazevAtributu="Hodnota atributu"/>

<NazevZnacky NazevAtributu="Hodnota atributu">Obsah značky</NazevZnacky>

XML – Značky / 4

<idjmenehoprostoru:NazevZnacky/>

ubuntu

XML – Entity

<

&totomesto;

&cokoliv;

ubuntu

XML - komentáře

```
<!-- komentář -->
```

XML – Instrukce pro zpracování

<?instrukce?>

<?identifikátor parametry/data?>

<?xml-stylesheet href="styl.css"
type="text/css"?>

XML – Struktura dokumentu

- Úvodní řádek
- (Definice jmenných prostorů)
- (Definice dalších typů)
- Tělo dokumentu

Úvodní řádek

```
<?xml version="1.0" encoding="UTF-8"?>
```

XML – Vybraná pravidla

- Kořenový prvek
- Nekřížení značek
- Uzavírání značek
- Uvození hodnot atributů
- Velikost znaků

Validace

Validace

- Well formed XML document
- Valid XML document

Well Formed

- Korektní vzhledem k syntaxi
- Párové značky
- Nekřížené značky
- Hodnoty atributů v uvozovkách (apostrofech)
- ...

Valid

- V souladu s deklarovanou strukturou
  - DTD
  - XML Schema
  - Relax NG
  - Schematron

Značkovací jazyky

DTD

ubuntu

Deklarace Typu Dokumentu - DTD

- Struktura dokumentu
- Uvádí se v hlavičce dokumentu (přímo - interní nebo odkazem - externí)
- Využívá se stále méně

Elementy – Prvky - Značky

<!ELEMENT název_prvku (seznam prvků, které může obsahovat s pravidly výskytů)>

<!ELEMENT okres (název, obec+, přednosta?)>

Prvky / Pravidla výskytu

Bez znaku Musí právě jednou

? Může ale nemusí právě jednou

+ Musí jednou nebo vícekrát

* Může ale nemusí vícekrát

Prvky / Neobsahující další

<!ELEMENT popis (#PCDATA)>

Oddělování definic prvků

- Čárkou – Všechny definice se musí využít
- Svislou čárou (Pipe |) - využije se jedna nebo druhá definice

<!ELEMENT obec (jméno, (starosta?|tajemník?), popis*)>

<!ELEMENT popis (#PCDATA|historie)>

Atributy

<!ATTLIST název_pvku
atribut1 typ povinnost|implicintní hodnota
atribut2 typ povinnost|implicintní hodnota
...
atributN typ povinnost|implicintní hodnota
>

Atributy - Typy

- CDATA - Libovolný text
- ID - Klíčový atribut
- IDREF – Cizí klíč
- NMTOKEN - Spojitý řetězec
- NMTOKENS - Seznam spojitých řetězců oddělených mezerou
- seznam možných hodnot - Uvádí se do závorek a oddělují se | Př.: Pohlaví(žena|muž) „žena“

Atributy - Příklad

<!ATTLIST okres

Kodok ID #REQUIRED

KodKraj IDREF #IMPLIED

Stav (strukturálně postižený|nepostižený) „nepostižený“

Nazok CDATA #IMPLIED>

Připojení DTD

- Interní
- Externí systémová
- Externí veřejná

Interní DTD

- Zapsaná přímo v dokumentu

```
<!DOCTYPE okres [
<!ELEMENT okres (název, obec+, přednosta?)>
<!ELEMENT obec (jméno, popis*, starosta?)>
]>
<okres>
  ...
</okres>
```

Systémová DTD

- Distribuovaná s XML dokumentem

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE okres SYSTEM "okres.dtd">
```

Veřejná DTD

- Dostupná přes Internet

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
  WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

XML Schema

XSD - Proč?

- Bohatější než DTD
- Rozšiřitelné
- Použití XML
- Datové typy
- Jmenné prostory

XSD – Datové typy

- Zjednodušují zejména:
  - Validaci
  - Práci s databází
  - Definování integritních omezení
  - Konverzi datových typů

XSD – využití XML

- Není nutné se učit další jazyk
- Je možné používat stejný editor a parser
- Je možné využívat XML DOM
- Je možné transformovat schémata s využitím XSL

XSD – Striktnost

- Např. datový typ date
  - Musí mít přesný tvar: YYYY-MM-DD

XSD - Struktura

```
<?xml version="1.0" encoding="UTF-
  8"?>

<xs:schema
  xmlns:xs="http://www.w3.org/2001/XM
  LSchema"
  elementFormDefault="qualified">

... deklarace

</xs:schema>
```

XML - Příklad

```
<?xml version="1.0"?>
<note>
<to>Petr</to>
<from>Jirka</from>
<heading>Připomínka</heading>
<body>Nezapomeň že teď už jsem
  vedoucím já!</body>
</note>
```

XML - DTD

```
<!ELEMENT note (to, from, heading,
  body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

XML - XSD

```
<xs:schema xmlns:"..."
targetNamespace="..." xmlns="..."
elementFormDefault="qualified">
<xs:element name="note">
    <xs:complexType>
      <xs:sequence>
<xs:element name="to" type="xs:string"/>
<xs:element name="from" type="xs:string"/>
<xs:element name="heading"
  type="xs:string"/>
<xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

DTD - Připojení

```
<?xml version="1.0"?>

<!DOCTYPE note SYSTEM
"http://www.w3schools.com/dtd/note.d
  td">

<note>
<to>Petr</to>
<from>Jirka</from>
<heading>Připomínka</heading>
<body>Nezapomeň že teď už jsem
  vedoucím já!</body>
</note>
```

XSD - Připojení

```
<?xml version="1.0"?>

<note
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchem
  a-instance"
xsi:schemaLocation="http://www.w3schools.c
  om note.xsd">

<to>Petr</to>
<from>Jirka</from>
<heading>Připomínka</heading>
<body>Nezapomeň že teď už jsem vedoucím já!
  </body>
</note>
```

XSD – Primitivní datové typy

xs:string

xs:double

xs:date YYYY-MM-DD

xs:integer

xs:decimal

xs:boolean

xs:time

XSD - Deklarace jednoduchého prvku

```
<xs:element name="xxx" type="yyy"/>
```

ubuntu

XSD - Deklarace jednoduchého prvku

```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

XSD - Deklarace jednoduchého prvku

```
<xs:element name="lastname"
  type="xs:string"/>

<xs:element name="age"
  type="xs:integer"/>

<xs:element name="dateborn"
  type="xs:date"/>
```

XSD - Implicitní hodnota

```
<xs:element name="color"
  type="xs:string" default="red"/>
```

ubuntu

XSD - Fixní hodnota

```
<xs:element name="color"
  type="xs:string" fixed="red"/>
```

ubuntu

XSD – Atributy

- **Jednoduché prvky nemohou mít atributy, pouze komplexní prvky mohou mít atributy**
- **Atributy se vždy deklarují jako jednoduché**

XSD - Atributy

```
<xs:attribute name="xxx"
  type="yyy"/>
```

ubuntu

XSD - Atributy

```
<lastname lang="EN">Smith</lastname>


<xs:attribute name="lang"
  type="xs:string"/>
```

XSD - Atributy

```
<xs:attribute name="lang"
  type="xs:string" default="EN"/>

<xs:attribute name="lang"
  type="xs:string" fixed="EN"/>

<xs:attribute name="lang"
  type="xs:string" use="required"/>
```

Omezení hodnot

XSD - Omezení rozsahem

```
<xs:element name="age">
<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="140"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

XSD - Omezení výčtem

```
<xs:element name="car">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="BMW"/>
    <xs:enumeration value="Škoda"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

XSD - Omezení výčtem

```
<xs:element name="car"
  type="carType"/>

<xs:simpleType name="carType">
   <xs:restriction base="xs:string">
     <xs:enumeration value="Audi"/>
     <xs:enumeration value="Golf"/>
     <xs:enumeration value="BMW"/>
   </xs:restriction>
</xs:simpleType>
```

XSD - Omezení vzorem

```
<xs:element name="letter">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-z]"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

XSD - Omezení vzorem

```
<xs:element name="initials">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-zA-Z][a-
  zA-Z][a-zA-Z]"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

XSD - Omezení vzorem

```
<xs:element name="year">

<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:pattern value="[0-9][0-9][0-
  9][0-9]"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

ubuntu

Omezení délky

XSD - Omezení přesné

```
<xs:element name="password">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:length value="8"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

XSD - Omezení rozsahem

```
<xs:element name="password">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:minLength value="5"/>
    <xs:maxLength value="8"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

ubuntu

Komplexní typy

XSD - Komplexní typy

- Prázdný
- Obsahující další prvky
- Obsahující text
- Obsahující text a další prvky

XSD – Komplexní typy - příklady

```
<vyrobek id="1345"/>
```

XSD – Komplexní typy - příklady

```
<zamestnanec>
<jmeno>Leoš</jmeno>
<prijmeni>Mareš</prijmeni>
</zamestnanec>
```

XSD – Komplexní typy - příklady

```
<zvire typ="brouk">Pytlík</zvire>
```

XSD – Komplexní typy - příklady

```
<popis>
Kolik višní tolik
  <ovoce>třešní</ovoce> ...
</popis>
```

XSD - Komplexní typ - definice

```
<xs:element name="zamestnanec"
  type="osoba"/>
<xs:element name="student"
  type="osoba"/>

<xs:complexType name="osoba">
   <xs:sequence>
      <xs:element name="jmeno"
  type="xs:string"/>
      <xs:element name="prijmeni"
  type="xs:string"/>
   </xs:sequence>
</xs:complexType>
```

ubuntu

XSD - Komplexní typ - dědičnost

```
<xs:complexType name="osobaroz">
  <xs:complexContent>
    <xs:extension base="osoba">
      <xs:sequence>
        <xs:element name="ulice"
type="xs:string"/>
        <xs:element name="obec"
type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

XSD - Pouze elementy

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname"
type="xs:string"/>
      <xs:element name="lastname"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD - Pouze text a atributy

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension
 base="xs:integer">
        <xs:attribute name="country"
 type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

ubuntu

Komplexní typy - Indikátory

XSD – Řazení

- All
- Sequence

XSD - All

```
<xs:element name="person">
  <xs:complexType>
    <xs:all>
      <xs:element name="firstname"
type="xs:string"/>
      <xs:element name="lastname"
type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

XSD - Sequence

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname"
type="xs:string"/>
      <xs:element name="lastname"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD – Výskyty

- Choice
- maxOccurs
- minOccurs

XSD - Choice

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee"
type="employee"/>
      <xs:element name="member"
type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

XSD - maxOccurs, minOccurs

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name"
 type="xs:string"/>
      <xs:element name="child_name"
 type="xs:string"
      maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Značkovací jazyky

GML – Geography Markup Language

ubuntu

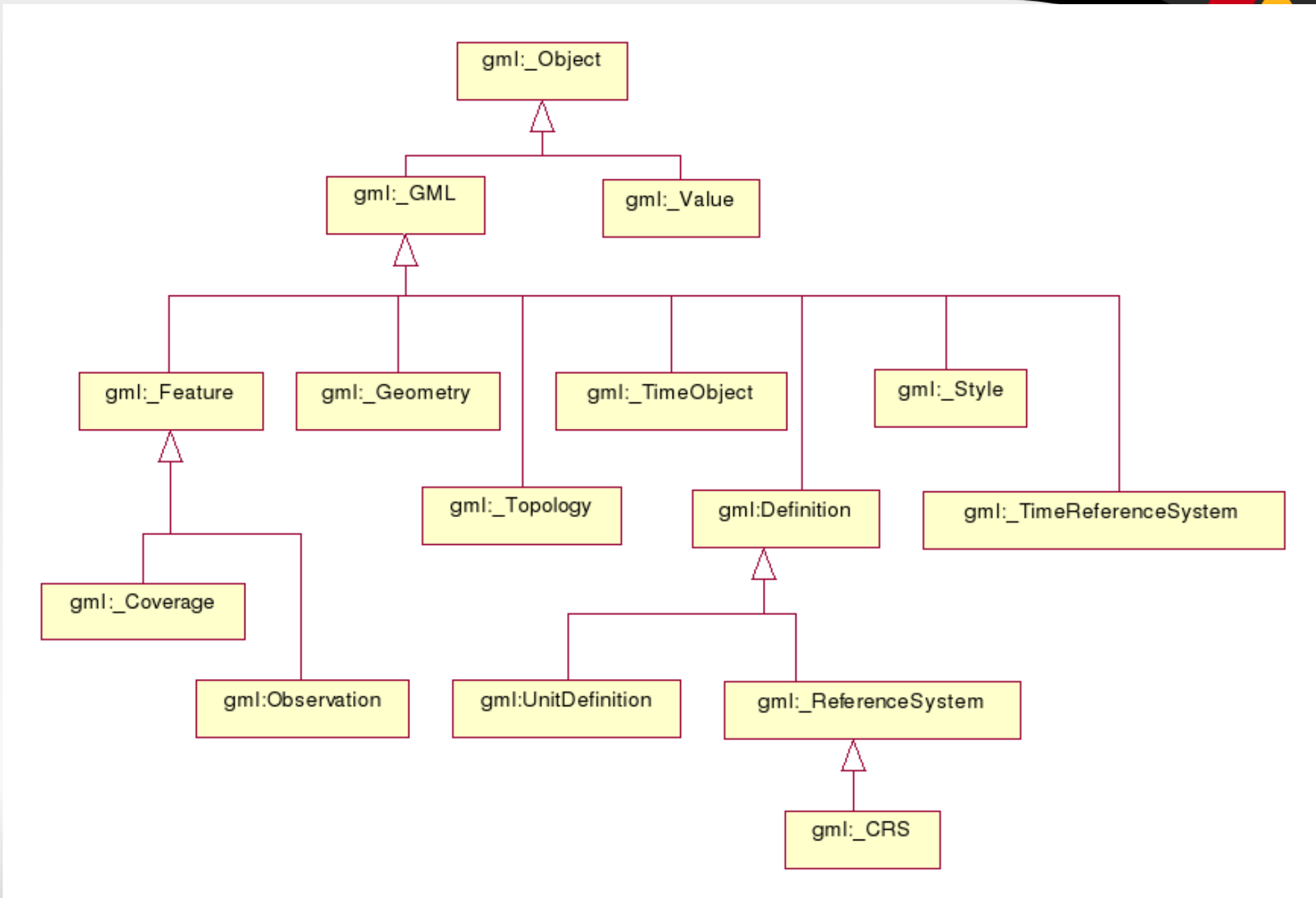
GML – Účel

- **Jazyk** pro modelování, přenos a ukládání **prostorových dat** včetně jejich prostorových a neprostorových vlastností
- Vytvořen pomocí jazyka **XML**
- Specifikace je definovaná **XML schématy**
- GML soubor=XML (textový) soubor
- Současná verze 3.1 = **ISO CD 19136** = Draft

GML - Schémata

- **GML v.2** tvoří 3 základní schémata:
  - **Geometry.xsd** – definuje geometrickou složku geoprvků
  - **Feature.xsd** – definuje hlavní model prvek-vlastnost. Rámec pro vytváření prvků
  - **Xlink.xsd** – definuje funkce pro odkazování
- **GML v.3** rozšířeno o **dalších 25** schémat
  - Kompatibilní s GML v.2

GML – Hierarchie



GML - Feature

- Vlastnosti (Properties)
  - Geometry (definováno v Geometry schema)
  - Topology
  - Temporal

ubuntu

GML – Geometry types

- **Primitives**
- **Complex – Composite**
- **Agregate - MultiPrimitive**

ubuntu

GML - Primitives

- **Point**
- **Curve**
- **Surface**

ubuntu

GML – Coordinate reference schema

- referenceSystems.xsd
- coordinateReferenceSystems.xsd
- datums.xsd
- coordinateSystems.xsd
- coordinateOperations.xsd
- dataQuality.xsd
- Založeno na ISO 19111

GML – Typy systémů

- Geocentric
- Temporal
- Engineering
- Image
- Derived
- Geographic
- Projected
- Vertical

GML – CRS - Jakost dat

- **dataQuality.xsd**
- **ISO 19115, ISO 19114**

ubuntu

GML – Topologie

- **ISO 19107**
- **gml:NodeType - Uzel**
- **gml:EdgeType – Hrana**
- **gml:FaceType – Plocha**
- **...**

ubuntu

GML – Čas a dynamika

- **ISO 19108:2002, ISO 8601, ISO 11404**
- **Čas pro geometrii, čas pro topologii, referenční systém**
- **Pro atributy i geoprvky**

ubuntu

GML - Čas a dynamika

```
<gml:track>
 <gml:MovingObjectStatus>
  <gml:validTime><gml:TimeInstant>
   <gml:timePosition>2005-11-
  28T13:00:00</gml:timePosition>
  </gml:TimeInstant></gml:validTime>
  <gml:location><gml:Point>
   <gml:pos>140. -35.</gml:pos>
  </gml:Point></gml:location>
  <gml:speed uom="#kph">12.</gml:speed>
  <gml:bearing>
   <gml:CompassPoint>SE</gml:CompassPoint>
  </gml:bearing>
 </gml:MovingObjectStatus>
```

GML - Čas a dynamika

```
 <gml:MovingObjectStatus>
  <gml:validTime><gml:TimeInstant>
   <gml:timePosition>2005-11-
  28T14:00:00</gml:timePosition>
  </gml:TimeInstant></gml:validTIme>
  <gml:location><gml:Point>
   <gml:pos>140.1 -34.9</gml:pos>
  </gml:Point></gml:location>
  <gml:speed uom="#kph">23.</gml:speed>
  <gml:bearing>
   <gml:CompassPoint>ESE</gml:CompassPoint>
  </gml:bearing>
 </gml:MovingObjectStatus>
</gml:track>
```

GML – Default styling

- **defaultStyle.xsd**
- **SMIL**

ubuntu

GML – Default styling

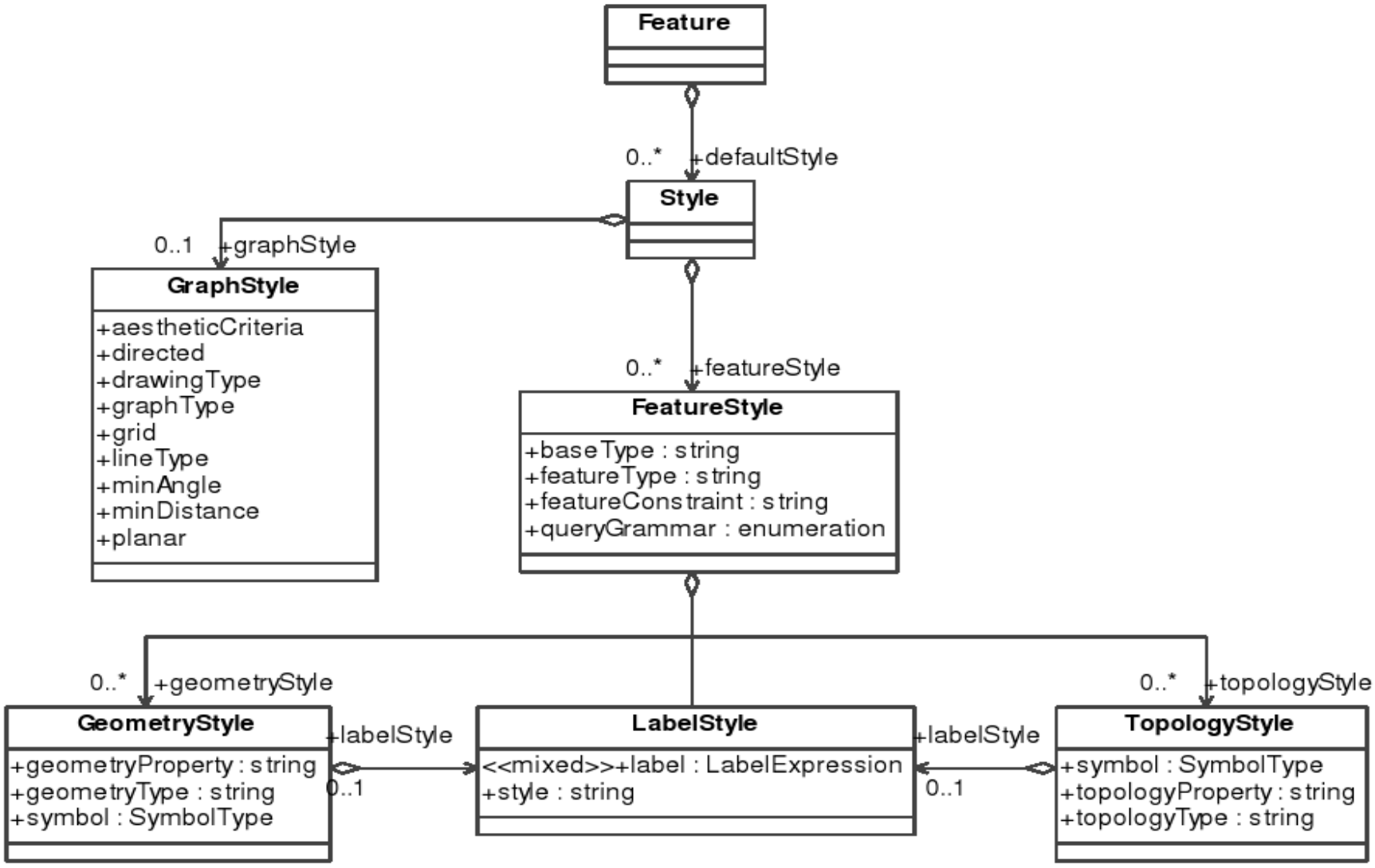


Figure 9 – The GML default styling containment model diagram

Značkovací jazyky

XSL (eXtensible Stylesheet Language)
XSLT (eXtensible Stylesheet Language Transformation)
XPath

ubuntu

XSL – Účel

- **Jazyk** pro transformaci XML dokumentů do jiných XML dokumentů nebo do jiných typů dokumentů (HTML, XHTML, VRML, PDF)
- Vytvořen pomocí jazyka **XML**

XSL - Části

- **XSLT** – jazyk pro transformaci XML dokumentů
- **XPath** – jazyk pro navigaci v XML dokumentech
- **XSL-FO** – jazyk pro formátování XML dokumentů

XSL – kořenový prvek

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XS
  L/Transform">
```

nebo

```
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XS
  L/Transform">
```

XSL - XML pro transformaci

```
<?xml version="1.0"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
...
</catalog>
```

XSL - XSL pro transformaci

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
    <tr><th>Title</th><th>Artist</th></tr>
    <xsl:for-each select="catalog/cd">
    <tr><td><xsl:value-of select="title"/></td>
       <td><xsl:value-of select="artist"/></td></tr>
    </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

XSL - Spojení XML a XSL

```
<?xml version="1.0" encoding="ISO-
  8859-1"?>
<?xml-stylesheet type="text/xsl"
  href="cdcatalog.xsl"?>
<catalog>
...
```

XSL - Template

- Šablona je spjata s prvkem dokumentu
- Atribut match určuje element s využitím jazyka XPath
- / znamená celý dokument – kořenový element

```
<xsl:template match="/">
...
</xsl:template>
```

XSL - value-of

- Hodnoty elementů se načítají s využitím značky value-of
- Atribut select identifikuje element s využitím XPath
- Cesta je relativní vůči nadřazenému elementu v našem případě je to kořenový element

```
<xsl:value-of
  select="catalog/cd/title"/>
```

XSL – for-each

- Průchod všemi určenými značkami
- Atribut select identifikuje značky s využitím XPath
- Cesta je relativní vůči nadřazenému elementu v našem případě je to kořenový element

```
<xsl:for-each select="catalog/cd">
...
</xsl:for-each>
```

XSL – filtrování for-each

- S využitím XPath je možné filtrovat značky

```
<xsl:for-each
  select="catalog/cd[artist='Bob
  Dylan']">
...
</xsl:for-each>
```

XSL - řazení

- Element sort zajistí setřídění výstupu for-each operace
- S využitím XPath je možné definovat způsob třídění

```
<xsl:for-each select="catalog/cd">
        <xsl:sort select="artist"/>
        ...
</xsl:for-each>
```

XSL – výběr

- Element if zajistí zpracování jen vybraných záznamů

```
<xsl:for-each select="catalog/cd">
    <xsl:if select="price &gt; 10">
    ...
    </xsl:if>
    <xsl:if select="price &lt; 10">
    ...
    </xsl:if>
</xsl:for-each>
```

XSL – výběr 2

- Element choose s elementy when a otherwise

```
<xsl:choose>
  <xsl:when test="price &gt; 10">
    <td bgcolor="#ff00ff">
    <xsl:value-of
  select="artist"/></td>
  </xsl:when>
  <xsl:otherwise>
     <td><xsl:value-of
  select="artist"/></td>
  </xsl:otherwise>
</xsl:choose>
```

XSL – kopírování elementů

- Pokud chcete pouze zkopírovat element do výstupu můžete použít dvě značky
- xsl:copy
- xsl:copy-of

XSL - copy

- Zkopíruje pouze element, bez dětí a atributů

```
<xsl:template match="message">
   <xsl:copy>
      <xsl:apply-templates/>
   </xsl:copy>
</xsl:template>
```

XSL - copy-of

- Zkopíruje element, děti i atributy

```
<xsl:template match="message">
  <xsl:copy-of select="body"/>
</xsl:template>
```

XSL - variable

- Je možné využívat proměnné

```
<xsl:variable name="header">
   <tr>
   <th>Element</th>
   <th>Description</th>
   </tr>
</xsl:variable>
<xsl:copy-of select="$header" />
```

Scalable Vector Graphics (SVG)

SVG – Účel

- **Jazyk** pro 2D vektorovou grafiku
- Založen na jazyce XML
- W3C recommendation
- Určen zejména pro oblast WWW
- Nachází uplatnění i v digitální kartografii

SVG – Části

- Grafické objekty
- Symboly
- Efekty s rastrovým obrazem
- Fonty
- Animace

ubuntu

SVG – Vykreslování

- Shapes, text
  - fill, stroke
  - solid color, gradients, patterns
- Raster
- Filtrace
- Maskování – clip, opacity

ubuntu

SVG – Struktura

- Element svg
- Element g
- Elementy rect, path, ...
- Element image
- Dědičnost
- ...

ubuntu

SVG - XSLT

```
<?xml version="1.0" standalone="no"?
  >
<svg width="10cm" height="5cm"
 xmlns="http://www.w3.org/2000/svg">
  <rect x="2cm" y="1cm" width="6cm"
 height="3cm"/>
</svg>
```

SVG - XSLT

```
<?xml version="1.0" standalone="no"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:svg="http://www.w3.org/2000/svg">
  <xsl:output
    method="xml"
    encoding="utf-8"
    doctype-public="-//W3C//DTD SVG 1.1//EN"
    doctype-system="http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"/>
  <!-- Add version to topmost 'svg' element -->
  <xsl:template match="/svg:svg">
    <xsl:copy>
      <xsl:copy-of select="@*"/>
      <xsl:attribute name="version">1.1</xsl:attribute>
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:template>
  <!-- Add styling to all 'rect' elements -->
  <xsl:template match="svg:rect">
    <xsl:copy>
      <xsl:copy-of select="@*"/>
      <xsl:attribute name="fill">red</xsl:attribute>
      <xsl:attribute name="stroke">blue</xsl:attribute>
      <xsl:attribute name="stroke-width">3</xsl:attribute>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

ubuntu

SVG - XSLT

```
<xsl:template match="svg:rect">
    <xsl:copy>
      <xsl:copy-of select="@*"/>
      <xsl:attribute
  name="fill">red</xsl:attribute>
      <xsl:attribute
  name="stroke">blue</xsl:attribute>
      <xsl:attribute name="stroke-
  width">3</xsl:attribute>
    </xsl:copy>
</xsl:template>
```

SVG - XSLT

```
<?xml version="1.0" encoding="utf-
  8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD
  SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1
  /DTD/svg11.dtd">
<svg width="10cm" height="5cm"
  version="1.1"
  xmlns="http://www.w3.org/2000/svg">
   <rect x="2cm" y="1cm" width="6cm"
  height="3cm" fill="red"
  stroke="blue" stroke-width="3"/>
</svg>
```

SVG - Souřadnice

- width, height
- viewBox (<min-x>, <min-y>, <width>, <height>)
- preserveAspectRatio

```
<svg width="453px" height="300px"
  viewBox="-5616 0 11039 11029"
```

SVG - Prostorové referenční systémy

- Interoperabilita
- Element metadata
  - reference na dokument
  - well-known identifier
  - úplně v dokumentu

ubuntu

SVG - Paths

- Cesty – mohou být vyplněny
- Polylinie, Eliptická, Beziérova, ...
- Zápis pomocí předpisu
- Prvky obecného předpisu
  - move to (M), line-to (L), close (z)

```
<path d="M 100 100 L 300 100 L 200 300 z"
        fill="red" stroke="blue" stroke-
  width="3" />
```

SVG – Základní tvary

- rect, circle, ellipse, line, polyline, polygon

SVG – Animace

- Elementy
- DOM
- SMIL

ubuntu

SVG – Animace

- Pohyb
- Barva
- Transform

ubuntu

Značkovací jazyky

VRML (Virtual Reality Modeling Language), X3D

ubuntu

VRML - Ukázka

```
#VRML V2.0 utf8
Transform {
  children
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0 1 0
        }
      }
      geometry Cylinder {
        height 0.1
        radius 0.5
      }
    }
}
```

ubuntu

X3D - Ukázka

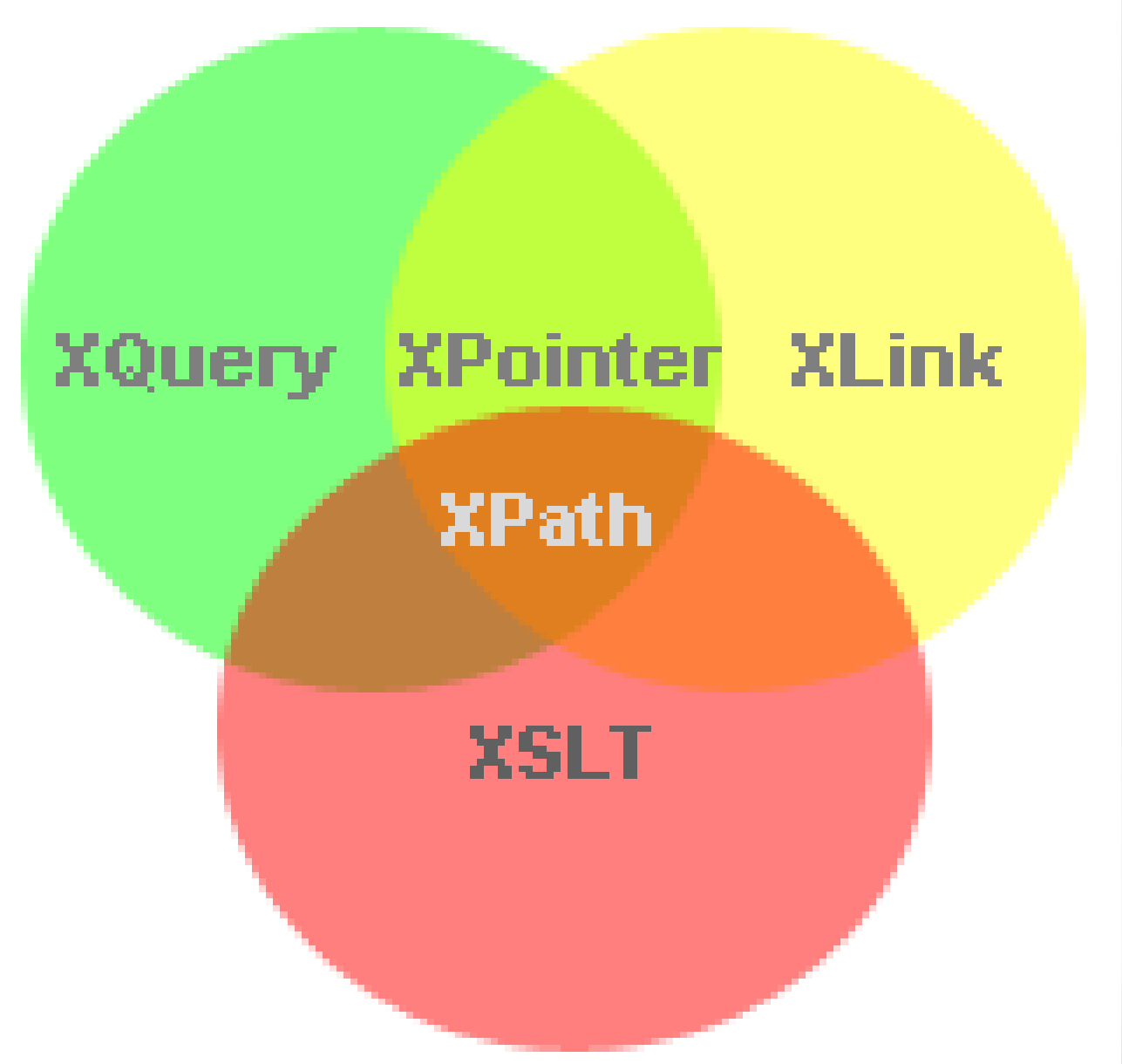
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D
  3.0//EN"
  "http://www.web3d.org/specifications/x3d
  -3.0.dtd">
<X3D profile="Immersive" version="2.0">
 <Scene>
  <Transform>
   <Shape>
    <Appearance>
     <Material diffuseColor="0 1 0"/>
    </Appearance>
    <Cylinder height="0.1" radius="0.5"/>
   </Shape>
  </Transform>
```

Značkovací jazyky

XPath, XLink, XQuery, XPointer,
XLinkTime, XForms

ubuntu

XPath – Vymezení



XPath – Výběr nodů

- `/` - výběr začíná od kořenového elementu
- <for-each select="//a">
- `//` - vybere nody, které splňují podmínku bez ohledu na to, kde v dokumentu se nacházejí
- `.` - vybere aktuální nod
- `..` - vybere rodičovský nod (parent node)
- `@` - vybere atribut

XPath – výběr - příklady

- **`/knihkupectvi`** – vybere nod knihkupectvi
- **`knihkupectvi/kniha`** – vybere všechny knihy, které jsou v knihkupectví
- **`//kniha`** – vybere všechny knihy v dokumentu bez ohledu na to, kde se nacházejí
- **`knihkupectvi//kniha`** – vybere všechny knihy, které jsou potomky knihkupectví, bez ohledu na vzdálenost od předka
- **`//@href`** – vybere všechny atributy href

XPath - Predikáty

```
/bookstore/book[1]
/bookstore/book[last()]
/bookstore/book[last()-1]
/bookstore/book[position()<3]
//title[@lang]
//title[@lang='eng']
/bookstore/book[price>35.00]
/bookstore/book[price>35.00]/title
```

XPath - Neznámé nody

```
*
@*

/bookstore/*
//*
//title[@*]
```

XPath – Funkce

- Více než sto vestavěných funkcí
- Skupiny nodů
- Řetězce
- Čísla
- ...

ubuntu

XPath - Funkce

```
count()
position()
concat()
starts-with()
contains()
substring()
sum()
...
```

XPath - Funkce

```
<xsl:for-each select="feature">
  <gml:featureMember>
    <prvek fid="{position()}">
...
    </prvek>
  </gml:featureMember>
</xsl:for-each>
```

XPath - Funkce

```
<xsl:when
  test="/METAIS['count(/SERVICE)'!
  ='0']">
```

XLink a XPointer

XLink – Účel

- Odkazy
- Jednoduché
- Rozšířené
- I mimo odkazované dokumenty

XPointer – Účel

- Odkazy na části dokumentů definovaných s využitím XPath

XPointer a XLink – Podpora

- Velice omezená podpora

XLink - simple

```
<homepage xlink:type="simple"
  xlink:href="http://www.w3schools.co
  m">Visit W3Schools</homepage>
```

XPointer - simple

```
<homepage xlink:type="simple"
  xlink:href="http://www.example.com/
  cdlist.xml#id('rock').child(5,item)
  "">Visit W3Schools</homepage>
```

ubuntu

XLink - jmenný prostor

```
<?xml version="1.0" encoding="ISO-
  8859-1"?>
<bookstore
  xmlns:xlink="http://www.w3.org/1999
  /xlink">
```

XLink - další

```
xlink:show="new"
xlink:show="embed"

xlink:actuate="onLoad"
xlink:actuate="onRequest"
```

XLink - extended

locator
arc
title
resource

ubuntu

Značkovací jazyky

Semantic Web, RDF, OWL

ubuntu

Semantic Web – Co to je

- Nová generace WWW
- Význam dat je srozumitelný i počítačům
- Programy mohou na základě obsahu odvozovat další informace

Semantic Web – Vyhledávání

- Ve vyhledávači zadám termín majoránka
- Najde mnoho naprosto nesourodých stránek
- Mě však může zajímat:
  - historie koření
  - recepty
  - mapa, kde se koření pěstuje
  - návod jak pěstovat
  - ...

Semantic Web – Kombinace lokálních dat

- Chci vytvořit kartodiagram produkce cukrové řepy v jednotlivých krajích ČR
- Sémantika dat zajistí správné propojení mých prostorových dat přes identifikátory, pokud to bude možné
  - csu:kraj_id
  - ssu:kraj_id

Semantic Web – Cíl

- Vytvořit platformu pro sdílení dat
- Data v relačních databázích, XML, proprietárních formátech
- Velká databáze, sdílená
- Jde o data a jejich význam

ubuntu

Semantic Web – Vize

- Data jsou popsána tak aby jim rozuměly i počítače
- Programoví agenti mohou daty procházet a vyhledávat v nich na základě významu
- Programoví agenti spolu mohou komunikovat a sdílet informace
- Programoví agenti se mohou učit rozumět pojmům z jiných doménových oblastí

Semantic Web – Struktura

- XML
- RDF
- Ontologie
- Inferenční mechanizmus (odvozování)
- Zabezpečení věrohodnosti (trust layer)
- Poučení uživatelé a vývojáři

RDF – Účel

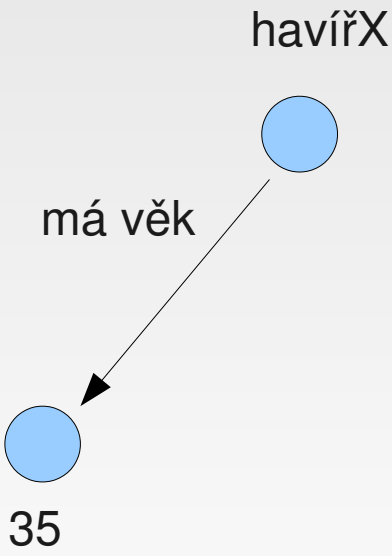
- Popis a provázání dokumentů (dat)
- RDF Triplet

RDF – Graf

- RDF je založen na grafech
- Subjekt, Objekt – nody
- Predicate – hrana
- Každý zápis v RDF představuje hranu a dva uzly

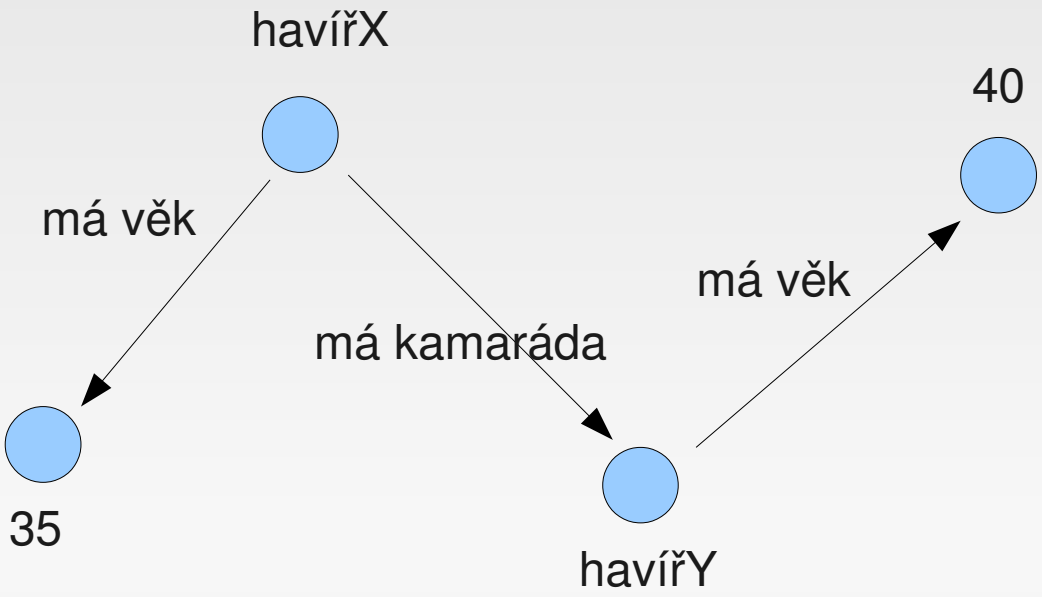
RDF – Graf

- Tvrzení (statement) ve formě tripletu
- Metadata
- XML



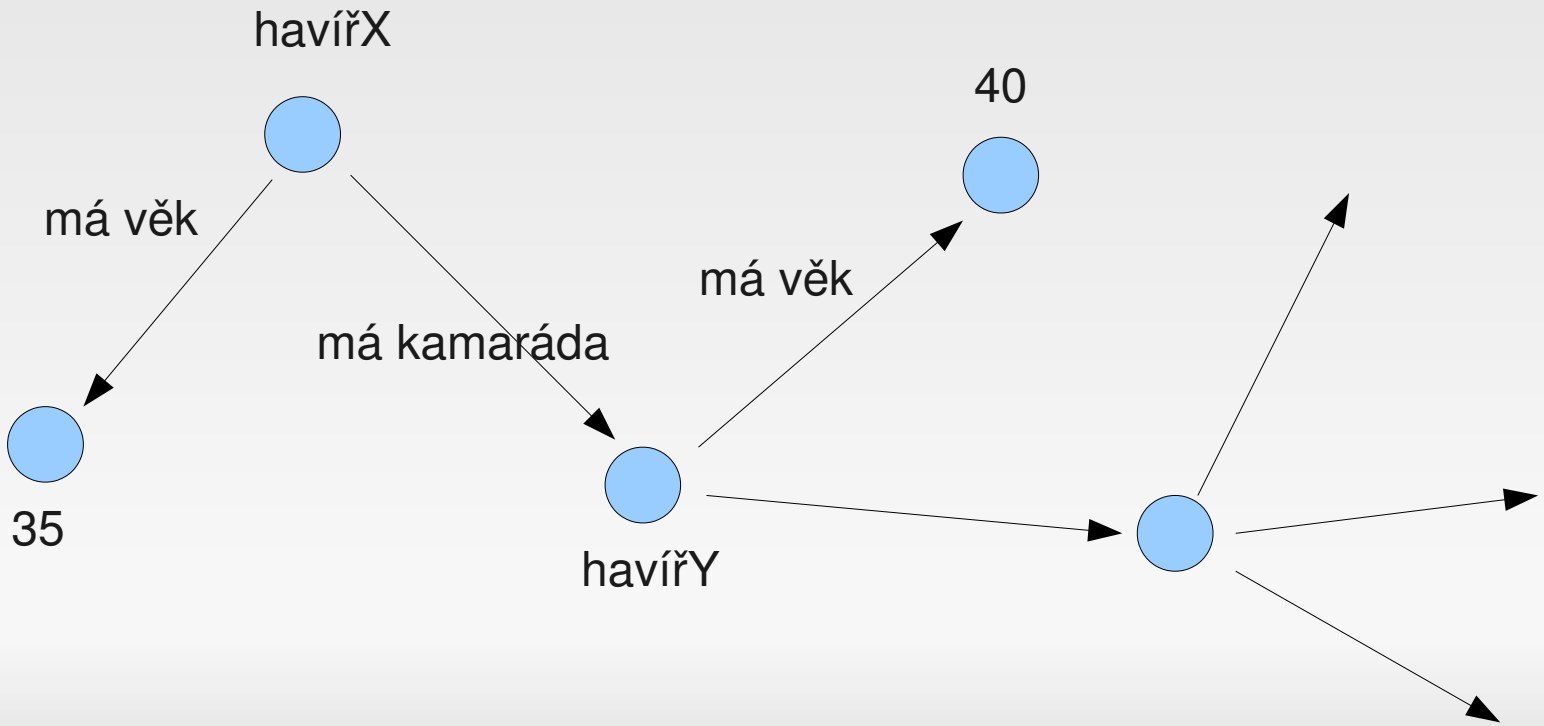
RDF – Graf

- Tvrzení (statement) ve formě tripletu
- Metadata
- XML



RDF – Graf

- Tvrzení (statement) ve formě tripletu
- Síť metadat



Ontologie

- Sdílený slovník popisující určitou oblast
- Popisuje typy objektů (třídy) a/nebo pojmy, jejich vlastnosti a vztahy mezi nimi

Ontologie

- explicitní specifikace konceptualizace [T. Gruber]
- formální specifikace sdílené konceptualizace [W. Borstem]

Ontologie – některé nástroje

- RDF
- RDF Schema
- OWL (Web Ontology Language)
- Prolog
- ...

Ontologie - RDF

```
<rdfs:Class rdf:about="Person"
  rdfs:label="Person">
     <rdfs:subClassOf
  rdf:resource="Human"/>
</rdfs:Class>
<rdf:Property rdf:about="hasSon">
     <rdfs:domain rdf:resource="Person"/>
     <rdfs:range rdf:resource="Person"/>
     <rdfs:subPropertyOf
  rdf:resource="hasRelative"/>
</rdf:Property>
```

ubuntu

Ontologie – odvozování

- Logické odvozování - Inference
- Součástí ontologie je soubor odvozovacích pravidel
- Podle těchto pravidel je možné odvodit informace, které nebyly implicitně uvedeny v metadatech

Ontologie – odvozování

- Pokud je havířX instancí třídy Person a jeho syn je PetrX

```
<Person>
   <Name>havířX</Name>
   <hasSon>PetrX</hasSon>
<Person>
```

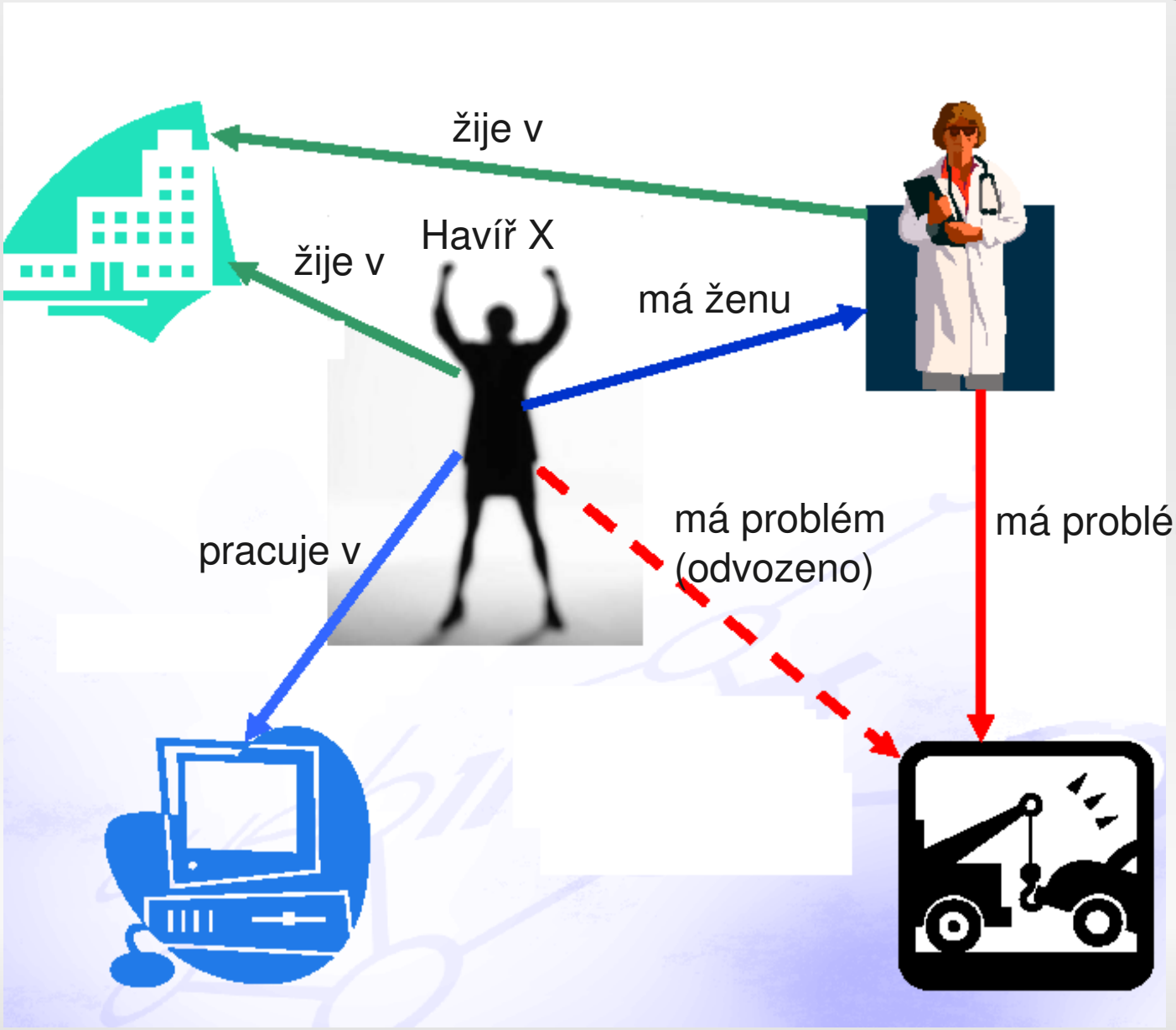
- Pak inferenční mechanismus dokáže zjistit, že havířX i PetrX jsou ze třídy *Human* a že jsou příbuzní (*hasRelative*)

Ontologie – ověřování

- Odvození nemusí být spolehlivé
- Mechanismus ověření odvození
- Digitální podpis (XML Security)

ubuntu

Ontologie – odvozování



OWL – Účel

- Rozšířit možnosti popisu ontologií
- Oproti RDF přidává např. kardinalitu vztahů
  - Určení bigamie (*učiněné šílenství*) :)

OWL – Třídy

- Identifikovaná třída
- Výčtem prvků
- Omezením vlastnosti
- Překryvem více tříd
- Sjednocením více tříd
- Doplňkem ke třídě

OWL - Výčtem prvků

```
<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Eurasia"/>
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#NorthAmerica"/>
    <owl:Thing rdf:about="#SouthAmerica"/>
    <owl:Thing rdf:about="#Australia"/>
    <owl:Thing rdf:about="#Antarctica"/>
  </owl:oneOf>
</owl:Class>
```

Semantic Web – Podmínky rozvoje

- Dokončení OWL
- Rozšíření povědomí
- Masové vytváření metadat a ontologií
- Vytvoření praktických aplikací

Semantic Web – Závěr

Nejzajímavější na sémantickém webu není to, co si dokážeme představit, že s ním můžeme dělat, ale to co si **představit nedokážeme**

Stejně jako jsme si před 10 lety nedokázali představit možnosti současného webu

Tim Bernes-Lee