

XSLT

eXtensible Stylesheet Language Transformation

Přednáška z předmětu Počítačová kartografie

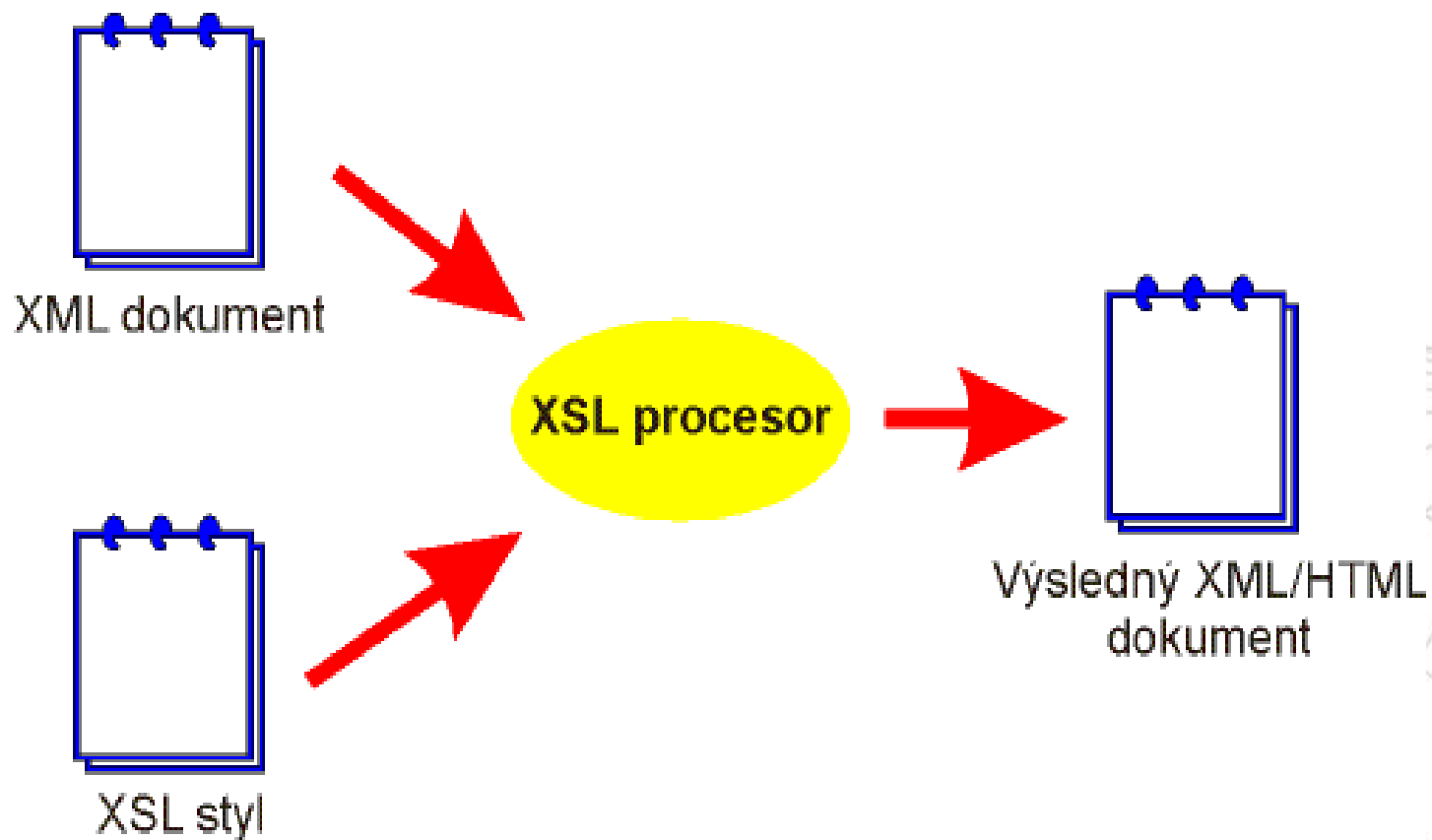
Otakar ČERBA
Západočeská univerzita v Plzni

XSLT

- Na pomezí mezi SQL (dotazovacími jazyky) a programovacími jazyky
- Obsahuje dotaz na data, ale také možnost úpravy (modifikace) těchto dat
- Nemá plnou funkčnost obojího, ale možnosti



Základní princip

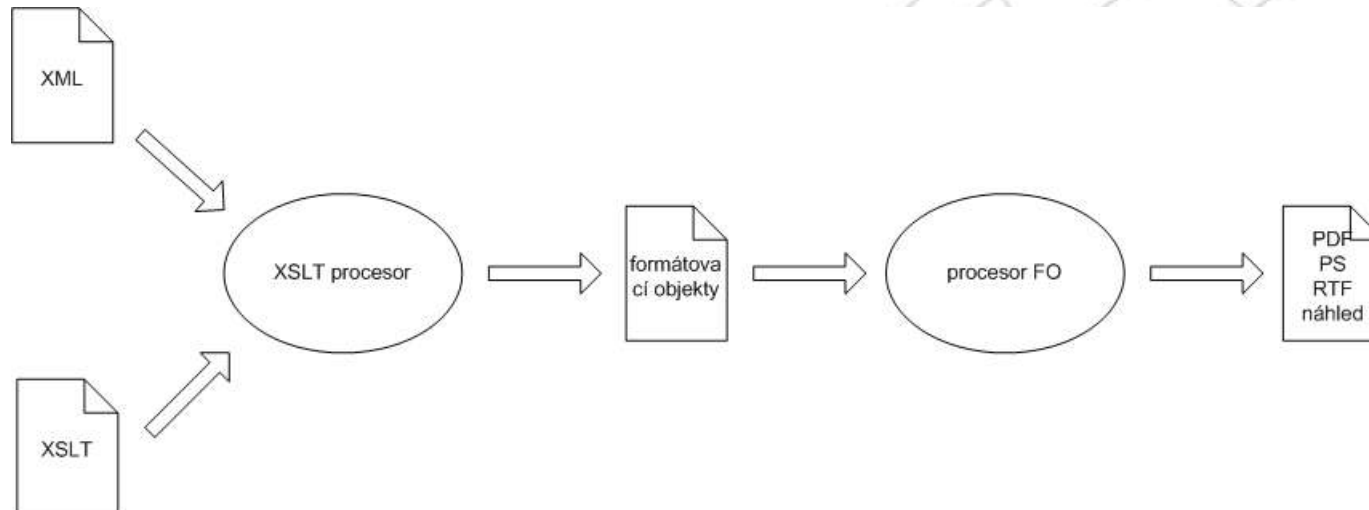
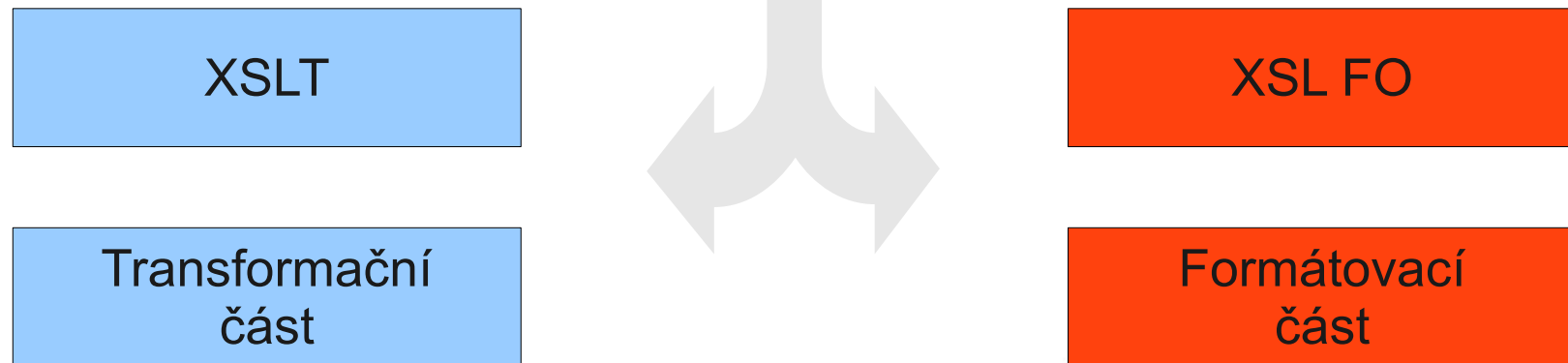


Zdroje informací

- Kosek, J. XSLT v příkladech
- Kosek, J. Přednášky předmětu IZI238 (MP3)
- Nič, M. XSLT 2.0 Tutorial
- W3C: XSLT Tutorial
- W3C Schools: The eXtensible Stylesheet Language Family
- Holman, G. K. What Is XSLT
- Tennakoon, W.T.M.S.B. Visualization of GML data using XSLT
- Extensible Stylesheet Language (XSL)

XSL

eXtensible Stylesheet Language



Transformační jazyky

- Kaskádové styly (CSS)
 - W3C standard
- Document Style Semantics and Specification Language (DSSSL)
 - ISO/IEC 10179/1996
- Formatted Output Specification Instance (FOSI)

CSS	Vlastnost	XML
Ano	Používání v HTML	Ne
Ano	Používání v XML	Ano
Ne	Transformační část	Ano
CSS	Syntaxe	XML

XSLT 2.0

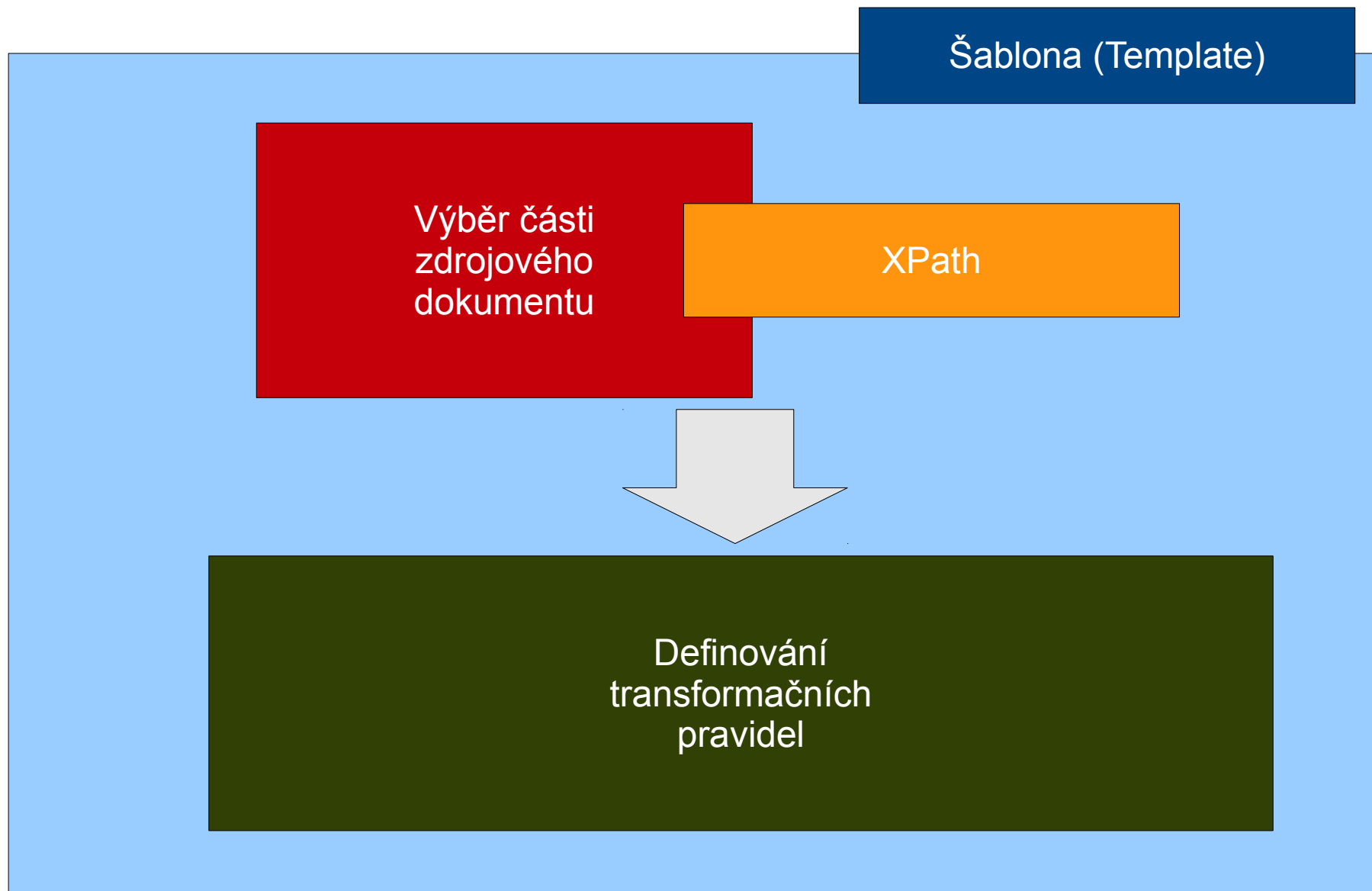
- XSLT 2.0 společně XPath 2.0 získaly 23.1.2007 (verze 1.0 - 16.11.1999) oficiální statut W3C Recommendation. Specifikace XSLT 2.0, XPath 2.0 a XQuery je tvořena celkem osmi dokumenty:
 1. XSL Transformations (XSLT)Version 2.0
 2. XML Path Language (XPath)2.0
 3. XQuery 1.0 and XPath 2.0 Functions and Operators
 4. XQuery 1.0 and XPath 2.0 Data Model (XDM)
 5. XQuery 1.0: An XML Query Language
 6. XSLT 2.0 and XQuery 1.0 Serialization
 7. XML Syntax for XQuery 1.0 (XQueryX)
 8. XQuery 1.0 and XPath 2.0 Formal Semantics

XSLT 2.0

- XSLT zásadně změnilo charakter. Od stylového jazyku (jakési dokonalejší verze kaskádových stylů) se posouvá spíše do oblasti programovacích jazyků, o čemž svědčí doplnění a zdokonalení práce s funkcemi, podmíněné výrazy apod.
- I have started to use XSLT 2.0 as my primary programming language (in combination with Python) and I am amazed by its power.

Miloslav Nič

Základní princip



XPath

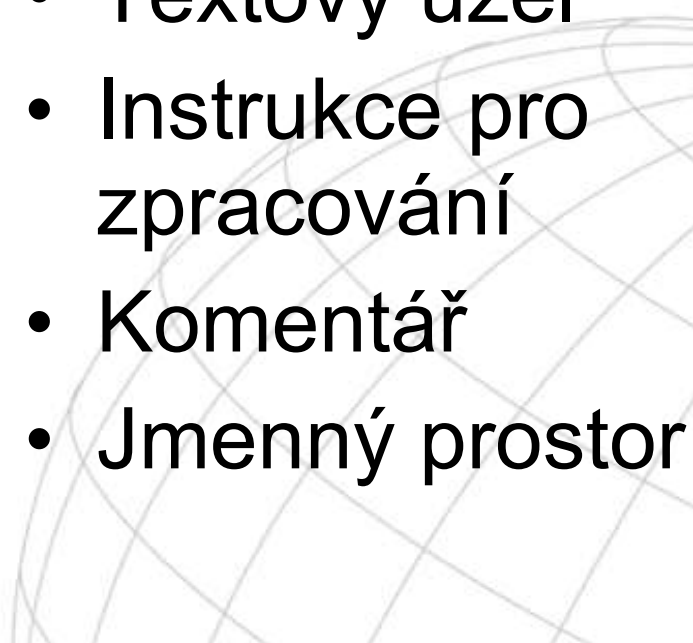
- Jednoduchý dotazovací jazyk, využívá se v aplikacích XSLT, XPointer, XML Schema (referenční integrita), XML Query (XQuery) XForms a dalších jazycích, proto tvoří samostatný standard.



XML Infoset

- XSLT procesor nepracuje přímo s fyzickou reprezentací XML dokumentu v souboru, ale operuje (dotazuje se) nad abstraktním modelem dokumentu, který si obvykle vytvoří v paměti.
- Model vychází se samostatné specifikace XML Infoset (XML Information Set).
- Celý dokument je reprezentován jako strom, který obsahuje uzly několika typů.

XML Infoset – typy uzlů

- Každý uzel ve stromové reprezentaci má kromě typu i svůj obsah. Ten se skládá z textu, který uzel obsahuje. Pro elementy, které nejsou listy, se text získá tak, že se spojí textové uzly, které jsou jeho potomky.
 - Kořenový uzel
 - Element
 - Atribut
 - Textový uzel
 - Instrukce pro zpracování
 - Komentář
 - Jmenný prostor
- 

XML Infoset - příklad

<Atlas>

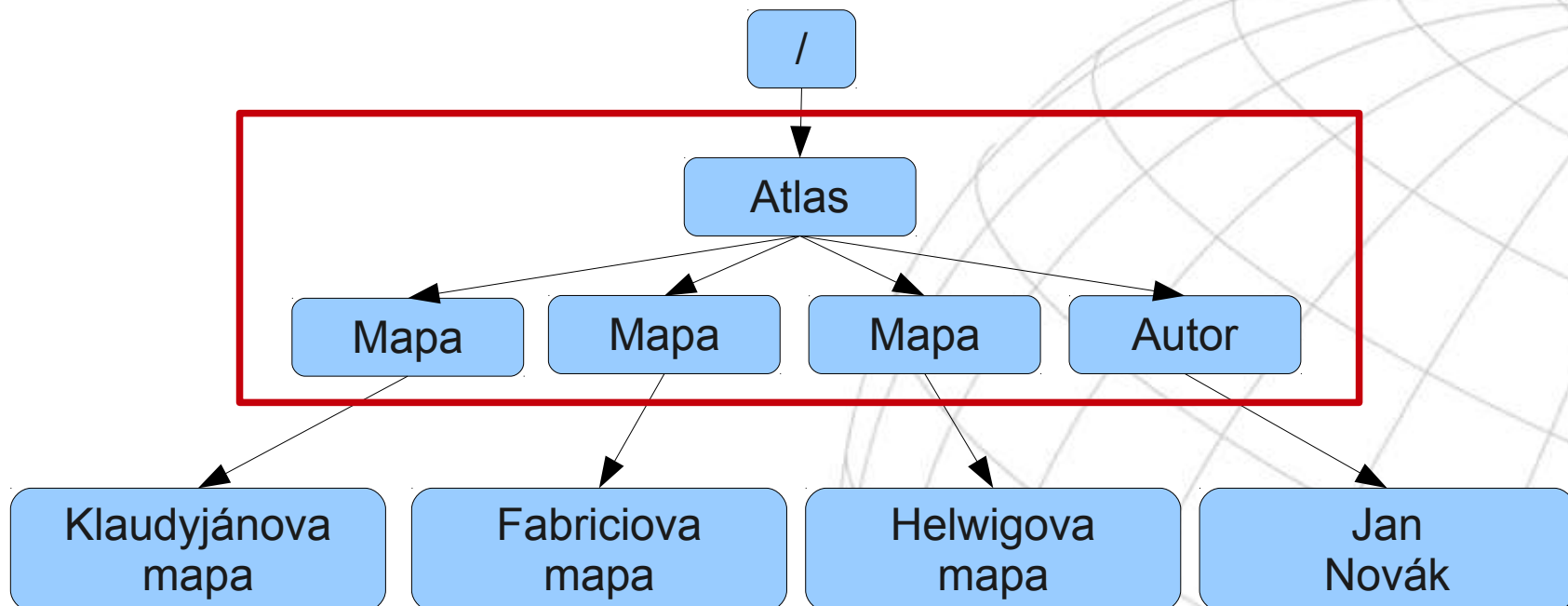
<Mapa>Klaudyjánova mapa</Mapa>

<Mapa>Fabriciova mapa</Mapa>

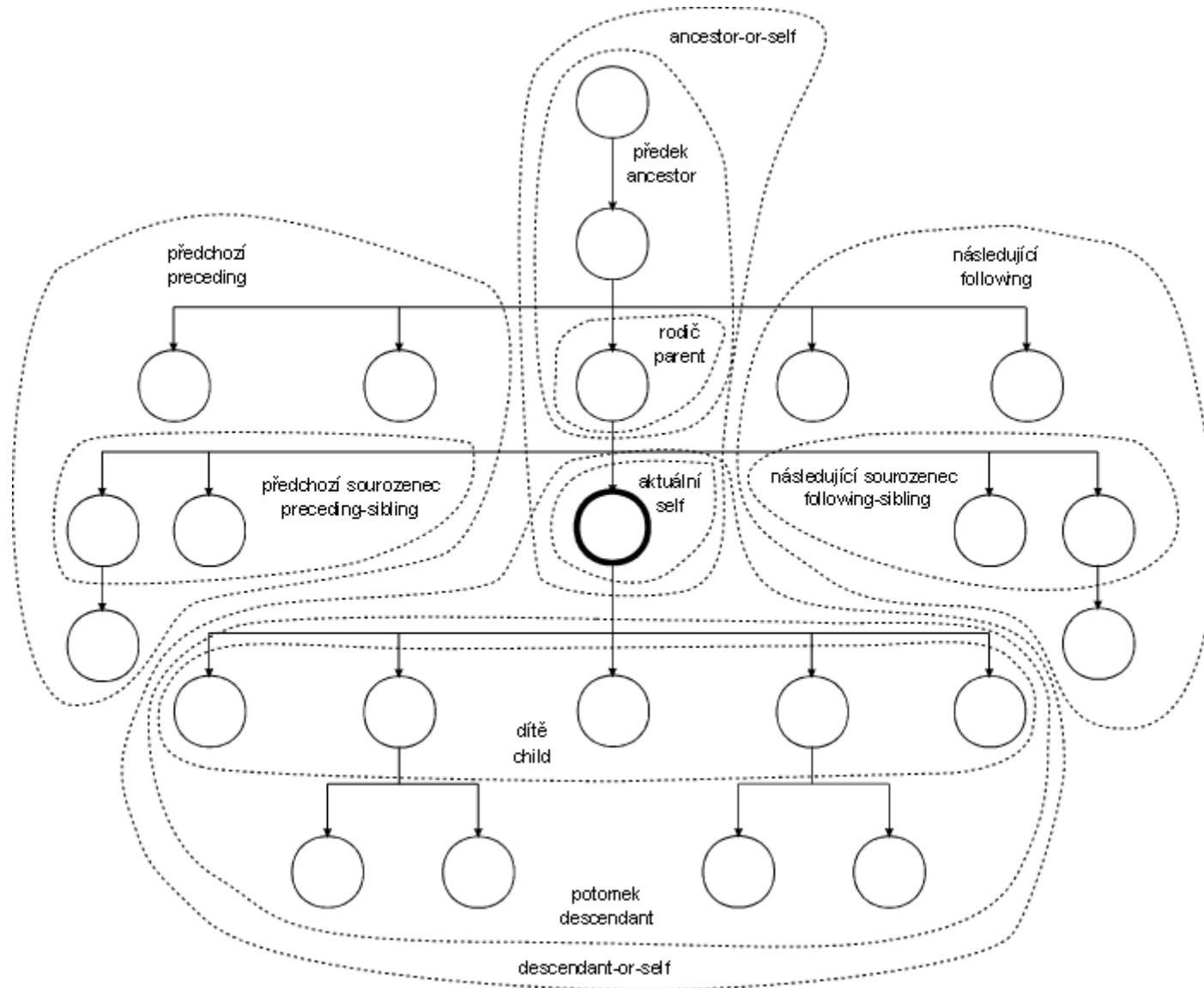
<Mapa>Helwigova mapa</Mapa>

<Autor>Jan Novák</Autor>

</Atlas>

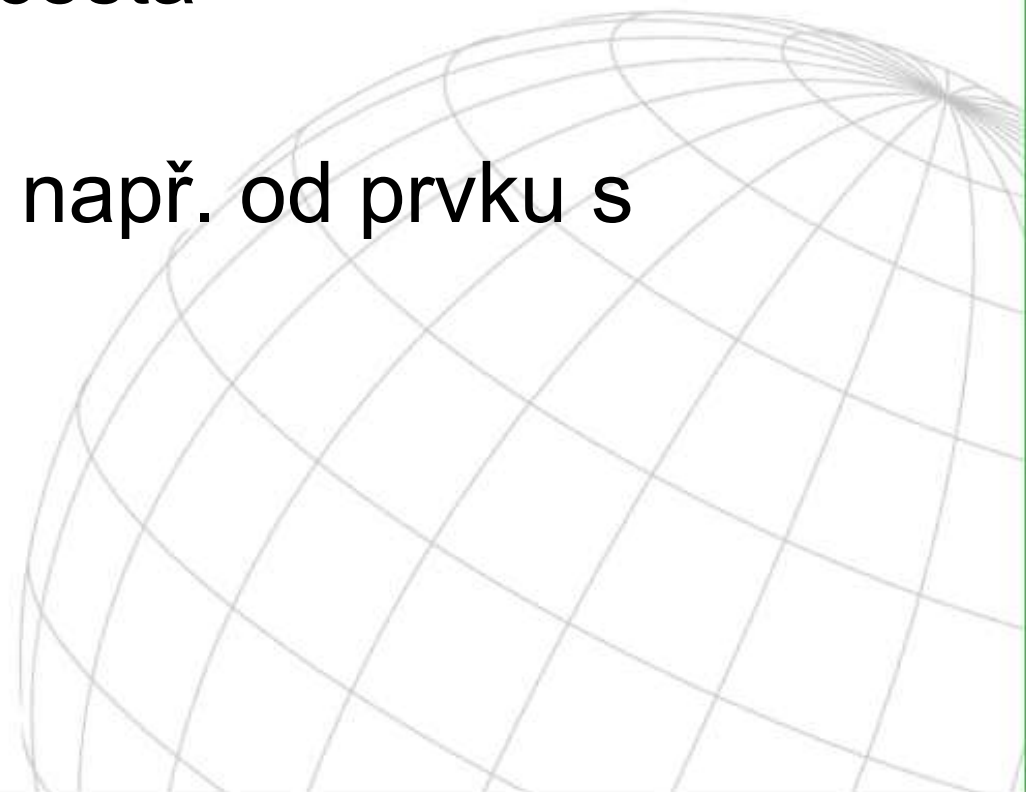


Vztahy mezi uzly



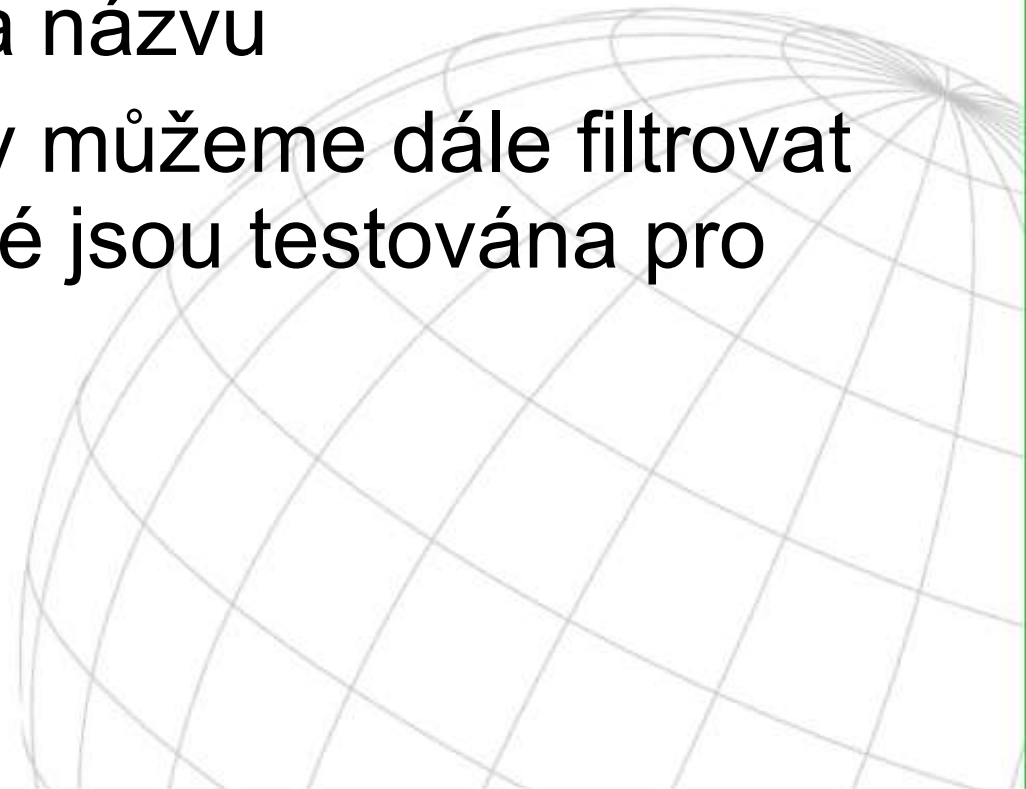
Začátek cesty (výběr uzlů)

- Od aktuálního uzlu – relativní cesta
 - Mapa
- Od kořene – absolutní cesta
 - /Mapa
- Od libovolného prvky – např. od prvku s konkrétním ID



Komponenty cesty

- Identifikátor osy – ten určuje, ve kterém směru se budeme od aktuálního uzlu pohybovat
- Test uzlu – umožňuje vybírat jen některé uzly na základě jejich typu a názvu
- Predikát – vybrané uzly můžeme dále filtrovat pomocí podmínek, které jsou testována pro každý uzel

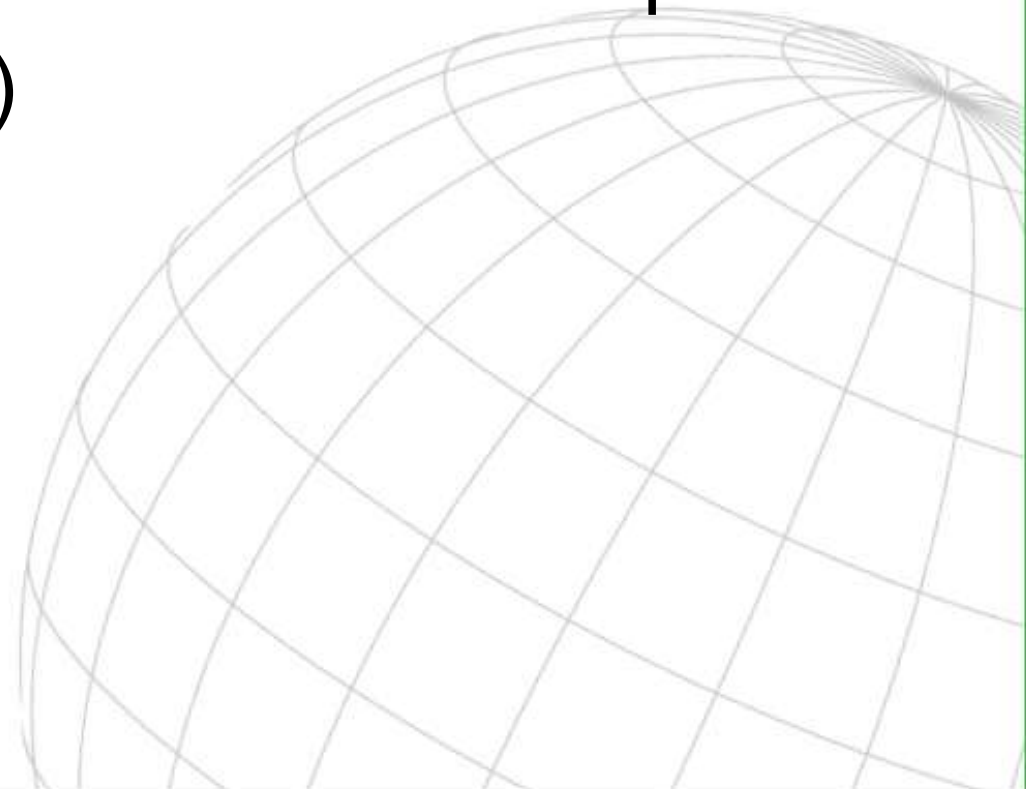


Identifikátory osy

- Celkem 13 druhů
- Při zápisu se za identifikátor osy doplňují dvě dvojtečky
 - Např. ancestor::
- ancestor-or-self
- parent
- child
- descendant
- following
- following-sibling
- preceding
- self
- attribute
- namespace

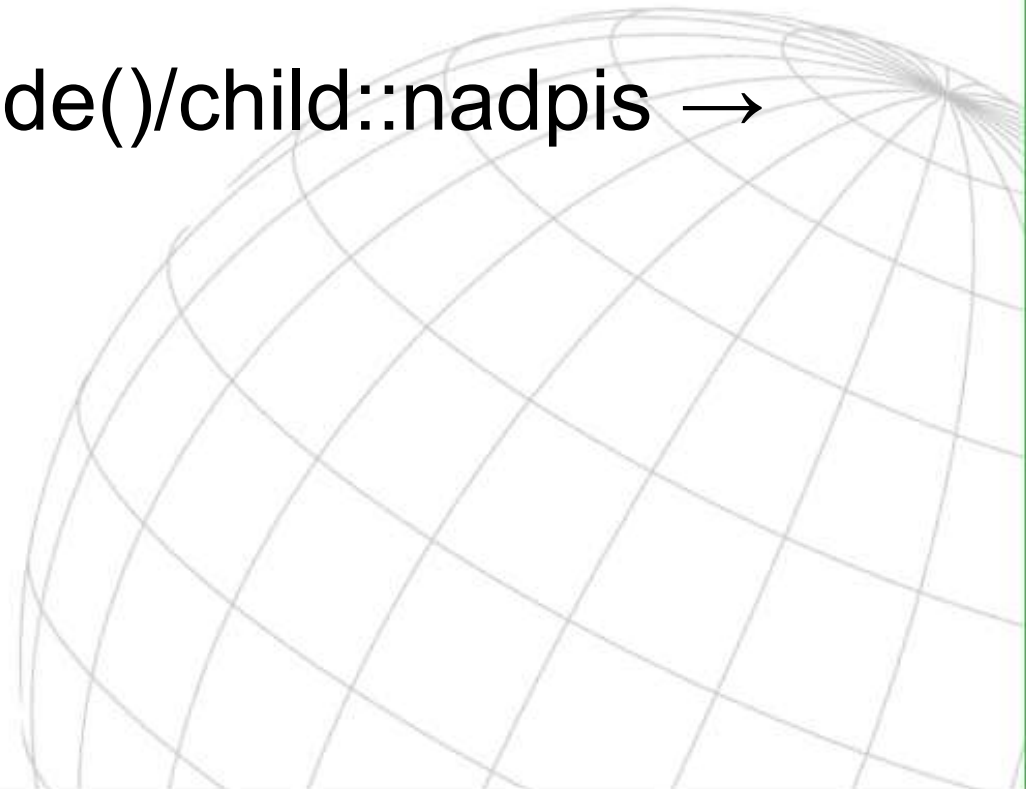
Test uzlu

- Konkrétní název uzlu
- Znak * – libovolný element
- prefix:* – libovolný element konkrétního prefixu
- processing-instruction()
- comment()
- text()
- node()



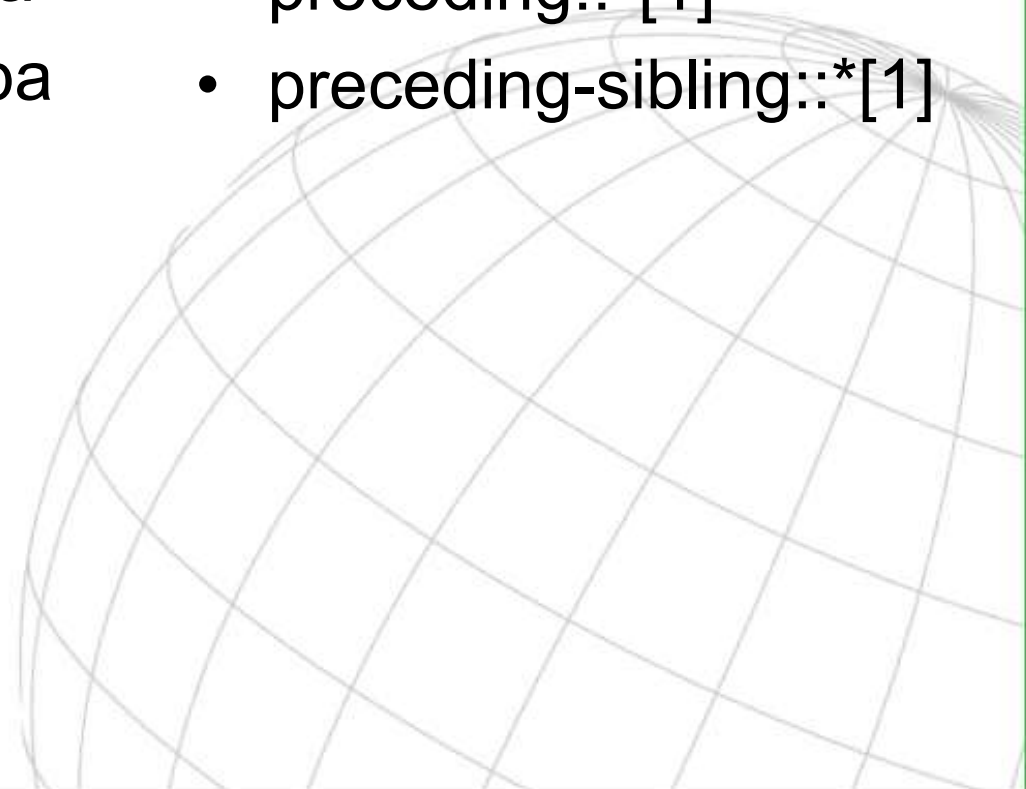
Zkrácené názvy uzlů

- `child::para` → `para`
- `attribute::id` → `@id`
- `self::node()` → `.`
- `parent::node()` → `..`
- `/descendant-or-self::node()/child::nadpis` → `//nadpis`



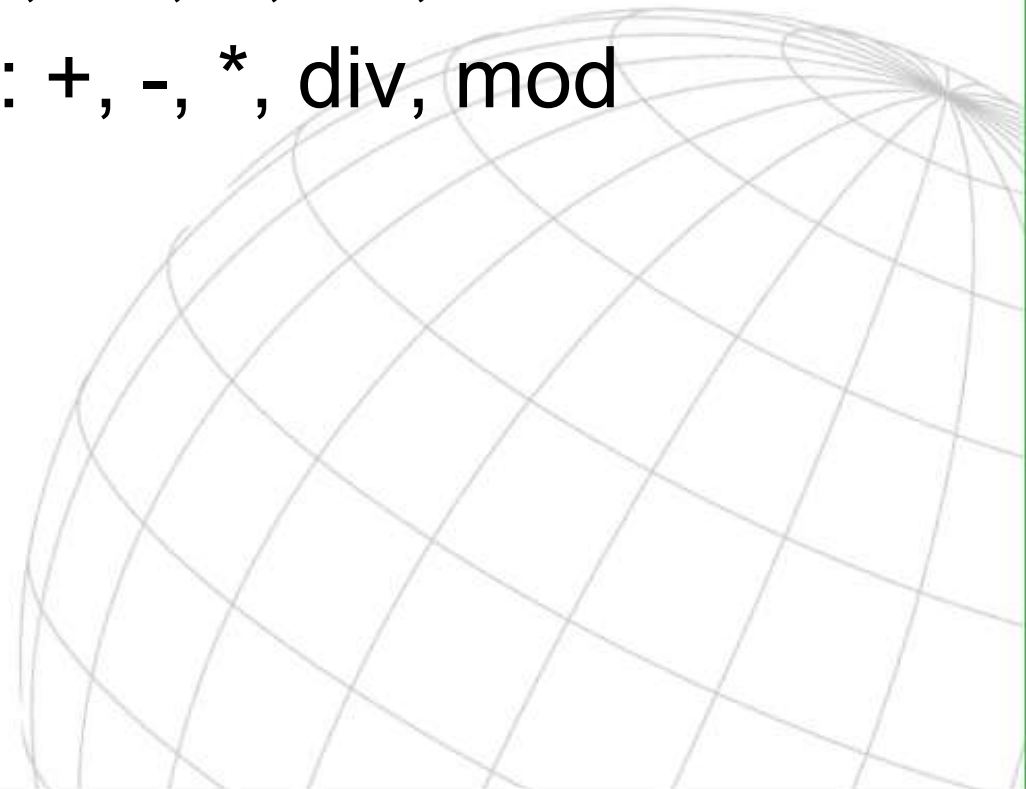
Predikáty - příklady

- Mapa
- /Mapa
- //Mapa
- *
- Mapa/*
- */Mapa
- @navez
- Mapa/@navez
- /*[@*]
- Mapa[1]
- Mapa[last()]
- Atlas/Sekce[3]
/Mapa[2]
- Atlas//Mapa
- //Atlas/Mapa
- .
- ./Mapa
- ..
- ../Mapa
- //Mapa[lang='cs']
- //Mapa[lang='cs']/Autor
- //Mapa[lang='cs'][4]
- //Mapa[4][lang='cs']
- preceding::*[1]
- preceding-sibling::*[1]

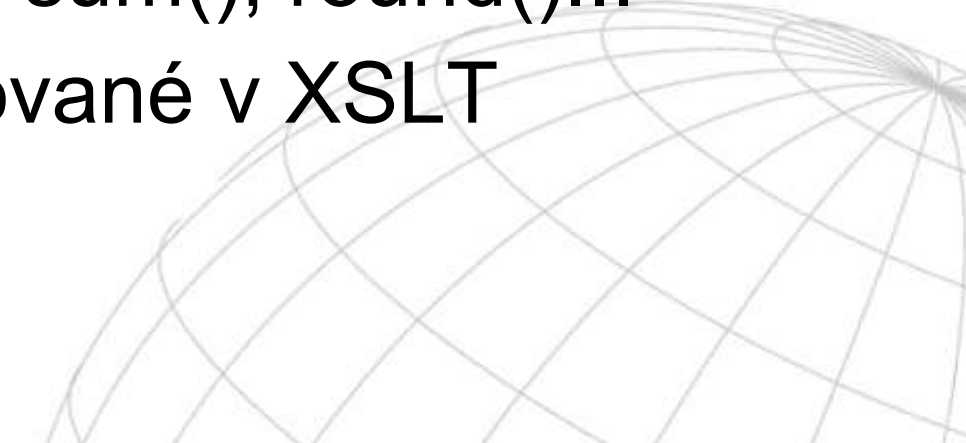


XPath operátory

- Sjednocení výrazů: |
- Logické spojky: and, or, not
- Relační operátory: =, !=, <=, <, >=, >
- Matematické operátory: +, -, *, div, mod



XPath funkce

- Práce s uzly: last(), position(), count()...
 - Řetězcové funkce: string(), concat(), substring()...
 - Logické funkce: boolean(), not(), true(), false()
 - Funkce pro práci s čísly: sum(), round()...
 - Rozšiřující funkce definované v XSLT
 - document()
 - generate-id()
 - format-number()...
- 

Připojení stylu

- Připojení v rámci XML dokumentu

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

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

```
<dokument>
```

```
...
```

```
</dokument>
```

- Připojení v rámci stylového procesoru

```
saxon -o Reserse.html Reserse.xml reserse_html.xsl
```



Hlavička stylu

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

```
<xsl:stylesheet
```

```
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

```
  version="1.0">
```

```
...
```

```
</xsl:stylesheet>
```



xsl:output

- Formát výstupního dokumentu
- method – xml, html, xhtml, text
- encoding – kódování výstupu
- + dalších 15 atributů (např. Standalone, version apod.)

```
<xsl:output method="xml" encoding="utf-8"/>
```

Šablona

Šablona (Template)

Výběr části
zdrojového
dokumentu

XPath

Definování
transformačních
pravidel

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

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/
XSL/Transform"
version="2.0">
<xsl:output method="xml"/>
```

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

</xsl:stylesheet>
```

xsl:apply-templates

```
<Kartografie>
  <Stat>France</Stat>
  <Stat>Spain</Stat>
</Kartografie>
```

```
<CartoGuide>
  <Country/>
  <Country/>
</CartoGuide>
```

```
<xsl:template match="/">
  <CartoGuide>
    <xsl:apply-templates
      select="Stat"/>
  </CartoGuide>
</xsl:template>
```

```
<xsl:template match="Stat">
  <Country/>
</xsl:template>
```

xsl:value-of

```
<Kartografie>
  <Stat>France</Stat>
  <Stat>Spain</Stat>
</Kartografie>
```

```
<CartoGuide>
<Country>France</Country>
<Country>Spain</Country>
</CartoGuide>
```

```
<xsl:template match="/">
  <CartoGuide>
    <xsl:apply-templates
      select="Stat"/>
  </CartoGuide>
</xsl:template>
```

```
<xsl:template match="Stat">
  <Country>
    <xsl:value-of select="."/>
  </Country>
</xsl:template>
```

xsl:value-of

```
<Kartografie>
  <Stat>France</Stat>
  <Stat>Spain</Stat>
</Kartografie>
```

```
<CartoGuide sum="2">
<Country>France</Country>
<Country>Spain</Country>
</CartoGuide>
```

```
<xsl:template match="/">
  <CartoGuide
    sum="{count(//Stat)}">
    <xsl:apply-templates
      select="Stat"/>
  </CartoGuide>
</xsl:template>
```

```
<xsl:template match="Stat">
  <Country>
    <xsl:value-of select="."/>
  </Country>
</xsl:template>
```

xsl:element

```
<Kartografie>  
  <Stat>France</Stat>  
  <Stat>Spain</Stat>  
</Kartografie>
```

```
<xsl:template match="/">  
  <CartoGuide>  
    <xsl:apply-templates  
      select="Stat"/>  
  </CartoGuide>  
</xsl:template>
```

```
<xsl:template match="Stat">  
  <xsl:element name="Stat{.}">  
    <xsl:value-of select="."/>  
  </xsl:element>  
</xsl:template>
```

```
<CartoGuide>  
<CountryFrance>France</CountryFrance>  
<CountrySpain>Spain</CountrySpain>  
</CartoGuide>
```

xsl:for-each

```
<Kartografie>
  <Stat>France</Stat>
  <Stat>Spain</Stat>
</Kartografie>
```

```
<xsl:template match="/">
  <CartoGuide>
    <xsl:for-each select="Stat">
      <Country>
        <xsl:value-of select="."/>
      </Country>
    </xsl:for-each>
  </CartoGuide>
</xsl:template>
```

```
<CartoGuide>
  <Country>France</Country>
  <Country>Spain</Country>
</CartoGuide>
```

Rozhodovací procesy

```
<xsl:if test="podmínka">
  příkazy
</xsl:if>
```

```
<xsl:choose>
  <xsl:when test="podmínka">
    příkazy
  </xsl:when>
  <xsl:when test="podmínka">
    příkazy
  </xsl:when>
  <xsl:when test="podmínka">
    příkazy
  </xsl:when>
  ...
  <xsl:otherwise>
    příkazy
  </xsl:otherwise>
</xsl:choose>
```


Další možnosti XSLT

- Parametry a proměnné
- Řazení a seskupování
- Používání klíčů
- Kombinování stylů
- Výstup do více souborů
- Více souborů na vstupu



Stylové procesory

- Saxon
 - Verze SA a B
 - Verze 6.x a 8.x
 - Saxon 9.0 (3.11.2007), Saxon 6.5.5
 - Verze Java a .NET
- Xalan
- XT
- libxslt/xsltproc
- MSXML
- .Net System.Xml



XSLT v kartografii

- Převod mezi různými datovými a metadatovými formáty (např. GML, JML, cGML, LandXML apod.)
- Tvorba vektorových digitálních map neboli převod prostorových dat do vektorových grafických formátů, například SVG nebo VML
- Vytváření jednoduchých dotazů a filtrů

XSLT 2.0 - sekvence

- Pro digitální kartografii (především pro generování map) je výhodná práce sekvencemi a textovými řetězci, které mohou představovat seznamy souřadnic (např. ve formátu GML nebo SVG). Otázkou je rychlost transformačních procesorů, které jsou většinou napsány v Javě, při zpracování takového objemu dat, který je v oblasti geoinformačních technologií běžný.

```
<aaa>
<bbb>b1</bbb>
<bbb>b2</bbb>
<bbb>b3</bbb>
</aaa>
```

```
<xsl:template match="/aaa">
  <yyy>
    <xsl:value-of select="bbb[3],bbb[1],bbb[2]"/>
  </yyy>
  <xxx>
    <xsl:value-of select="bbb[2],12,'c',bbb[1]"/>
  </xxx>
</xsl:template>
```

```
<yyy>b3 b1 b2</yyy>
<xxx>b2 12 c b1</xxx>
```

XSLT 2.0 - prohledávání

- Prohledávání a rozřazování rozsáhlých dokumentů obsahující prostorová data s velkým počtem atributů zjednoduší a zřejmě také zrychlí používání klíčů a možnost seskupování dat na základě zadaného výrazu (velice jednoduše se budou například řadit obce na základě příslušnosti k obci s rozšířenou působností).

```
<aaa>
<bbb v="a2"/> <ccc v="1"/>
<ddd v="1"/> <eee v="a2"/>
<fff v="1"/> <ggg v="zz3"/>
</aaa>
```

```
<xsl:template match="/aaa">
<xsl:for-each-group select="*" group-
by="@v">
<sss><xsl:apply-templates
select="."/></sss>
<ggg id="{current-grouping-key()}">
<xsl:apply-templates select="current-
group()"/>
</ggg>
</xsl:for-each-group>
</xsl:template>
```

```
<xsl:template match="*">
<xsl:copy-of select="."/>
</xsl:template>
```

```
<sss>
<bbb v="a2"/>
</sss>
<ggg id="a2">
<bbb v="a2"/>
<eee v="a2"/>
</ggg>
<sss>
<ccc v="1"/>
</sss>
<ggg id="1">
<ccc v="1"/>
<ddd v="1"/>
<fff v="1"/>
</ggg>
<sss>
<ggg v="zz3"/>
</sss>
<ggg id="zz3">
<ggg v="zz3"/>
</ggg>
```

XSLT 2.0 - matematika

- XSLT 2.0 integrovala řadu funkcí EXSLT, které jsou při tvorbě digitálních map nezbytné. Například se jedná o matematické funkce (součet, průměr, maximum, minimum) používané při tvorbě grafů a diagramů při generování kartodiagramů nebo při generování intervalů stupnic při generování kartogramů.


```
<aaa>
<a>5</a>
<a>3</a>
<a>7</a>
<a>2</a>
</aaa>
```

```
<bbb>99</bbb>
<ccc>7</ccc>
<ddd>2</ddd>
<eee/>
```

```
<xsl:template match="/aaa">
<bbb>
<xsl:value-of select="max(1 to 99)"/>
</bbb>
<ccc>
<xsl:value-of select="max(a)"/>
</ccc>
<ddd>
<xsl:value-of select="min(a)"/>
</ddd>
<eee>
<xsl:value-of select="max(b)"/>
</eee>
</xsl:template>
```

XSLT 2.0 - schémata

- Práce s datovými typy XML Schema, které jsou přebírány i do dalších aplikací (např. jazyky RELAX NG, OWL) je důležitá z hlediska tvorby obecného sémantického dokumentu a také snažší kontrole správnosti dokumentu (zabrání se tak například používání textových řetězců místo čísel apod.).



XSLT 2.0 – regulární výrazy

- Práce s regulárními výrazy patří mezi další výhody druhé verze XSLT. Například v SVG souborech půjde odstranit vysoké hodnoty jednotlivých souřadnic (dojde ke zmenšení velikosti souborů), „odříznutá“ hodnota bude do souboru vrácena pouze jednou ve formě translační transformace.



Atlas mezinárodních vztahů

