Technologie XML

Úvod do XML

Jiří Měska (jiri.meska@gmail.com)

MIB008, Datové a procesní modely

MFF UK Praha, 2020

Vraťme se k historii

Co nás vede k tomu, abychom se věnovali strukturovaným souborům pro přenos dat, jako je XML, JSON?

Abychom na to odpověděli, je třeba se vrátit trochu do historie, ale ne zas tak daleko. Vše co se týká IT není zas tak dávná historie (kromě hraběnky Ady Lovelace, ale to již jistě probral pan Kamenický)

Počítač je pro každého

Na trhu se objevují osobní počítače. Má je skoro každý kdo jde s dobou, krok dopředu.

Operační systém DOS (Microsoft) je primitivní (z hlediska klasických střediskových počítačů), krok vzad.

Rozvíjí se však komunikace mezi počítači, krok vpřed.

Rozvíjí se grafické komunikační rozhraní člověk počítač (do té doby doména drahých počítačových stanic), krok vpřed.

A lide?

Svět se propojuje

Na světě je mnoho komunikačních protokolů, chytřejší, spolehlivější, méně chytré, méně spolehlivé, svět ale propojí armádní projekt z konce 60 let (z dílny DARPA) TCP/IP, velký skok dopředu.

Nejprve univerzity, ale brzy skoro každý má svoji emailovou adresu, vzdálenost mezi lidmi se zkracuje. A pracovat lze asynchronně (email komunikace je asynchronní, telefon je synchronní). V Česku je telefon dostupný pro každého!

Jakýsi Tim Berners-Lee, působící v CERNu v roce 1990 navrhl protokol HTTP a jazyk HTML. Spustil první webový server a napsal první webový prohlížeč. Revoluční myšlenka byla ve vzájemných odkazech mezi webovými stránkami na různých webových serverech přes http protokol. Vzniká internet.

A samozřejmě Microsoft přichází z prvními Windows a ovládne svět operačních systémů na osobních počítačích. Kancelář i tiskárnu máme na stole.

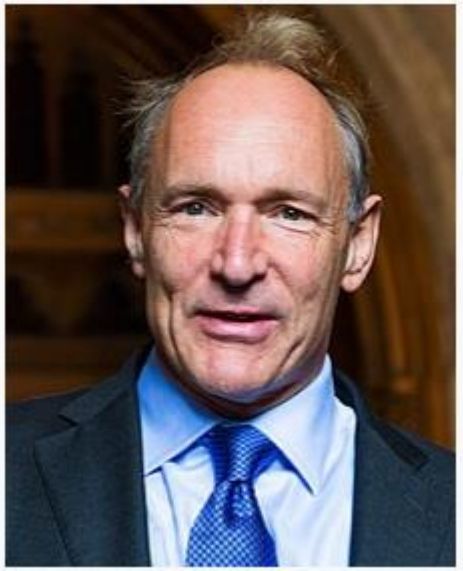
Ne každý je fanda Microsoftu. Proti Microsoftu se postaví open software, rozvíjí se UNIX. Vzniká konkurence.

90 léta jsou rájem pro programátory, líhní nových myšlenek.

A lidé?

A lidé?

Sir Timothy John Berners-Lee



James Gosling, the creator of Java, in 2008

Internet zdroj služeb

A lidé?

Vedle C a C++ se prosazuje Java. Sun Microsystems released the first public implementation as Java 1.0 in 1996.

Průvodcem internetem se nestávají katalogy, jak bylo zpočátku zamýšleno, ale vyhledávače. A máme zde Google.

V Čechách vznikl seznam.cz. Jeho mapy.cz jsou bezkonkurenční ☺.

Internet jako obchodní platformu objevuje Amazon.

Samostatnou kapitolou jsou mobilní telefony, ale z těch se stanou počítače až o dekádu později.

Unikátním projektem je GPS, které výrazným způsobem propojí virtuální svět a skutečný. A za to mohou chytré telefony.

Nejen lidé brouzdají internetem, ale i programy. Jak se ale v té změti informace vyznat? Je potřeba do toho zavést trochu pořádku.

Koncem 90 let se objevuje standard XML (Standard XML 1.0 je z roku 1999, 1.1 z roku 2004), jazyk pro výměnu dat/informací.

Navazuje standard webových služeb (WSDL), postavený na XML a umožňující publikovat služby (funkcionalitu) na internetu.

Později začne s rozvojem JavaScriptu konkurovat XML formát JSON.

A XML/JSON to je náš další program!

James Gosling, the creator of Java, in 2008

Larry Page and Sergey Brin in 2003

Čím se budeme zabývat

Dnes

- Co je to XML (Úvod do XML)

- Co je to dobře formátované XML a co je validní XML. To prvé je o syntaxi XML dokumentů, to druhé popisuje určitou třídu XML dokumentů

- Jak publikovat XML dokumenty z databáze prostředky SQL (SQL/XML)

Příště si řekneme

- Jak XML do databáze ukládat a jak s ním v databázi pracovat

- Něco málo o JSON formátu a možnostem SQL s ním pracovat

Formáty souborů pro přenos dat z/do databáze

Při exportu tabulky Obchodníků z Postgresu příkazem

```
copy  Obchodnik to 'c:/Temp/Obchodnik.txt';
```

jsme se setkali s následujícím CSV (comma-separrated values) souborem:

```
1002    \N      Novák    400160010.2  36
1001    1002    Charvát  400160010.1  41
1003    1002    Ryšavý   400360030.1  18
1004    1002    Pilný    400460040.1  \N
1005    \N      Horvát   400560090.15 \N
1006    1002    Ujen     400660060.2  51
1007    1006    Nhac     400760060.1  \N
…
```

Soubor bychom mohli použít k zálohování, přenosu nebo dalšímu zpracování dat z naší databáze. Ale nejspíše mu rozumí pouze naše databáze.

CSV soubory snadno načteme tabulkovými kalkulátory, např. Microsoft Excel

Je soubor čitelný? Co potřebujeme vědět, abychom mohli textový soubor přečíst?

Je soubor srozumitelný? Co potřebuje vědět, abychom našemu souboru rozuměli?

Zpracování/Interpretace souboru

Se soubory dat nemůžeme rozumně pracovat, aniž bychom něco nevěděli syntaxi a sémantice dat v něm uložených. Potřebuje nějaké metainformace. A míra této informace určuje co s tímto souborem můžeme dělat.

Metainformací z hlediska např. DOS/Windows vzhledem k souboru je například jméno, extension, datum vytvoření. Jméno souboru je v těchto OS rozděleno na jméno a příponu (extension) (rozdělovacím znaménkem je poslední tečka).

Tato informace je pro OS dostačující k tomu, aby uměl rozumně zobrazovat adresářové struktury a aby uměl spouštět programy, které s nimi umí pracovat. OS si mapuje přípony na programy.

Co jakou metainformaci potřebujeme k zobrazení textového souboru? Potřebujeme znát kódové tabulku v jaké je text uložen(Windows-1250, UTF8, UTF16).

Pokud bychom chtěli přečíst soubor .doc - obvykle soubor Word, k tomu potřebujeme znát jeho strukturu, a ta je hodně komplikovaná …

Zajímají-li nás soubory pro čitelné uložení dat z databáze, potom klasickým formátem je csv soubor, kde oddělovač polí v řádce je mezera (nebo čárka, středník) Typický csv soubor je obrazem databázové tabulky, ale i listu z Excelu.

Co je například metainformace k fotografii?

Jaký je rozdíl mezi bitmapou a jpeg?

Formáty souborů pro přenos dat

S rozvojem internetu a publikací dat na internetu vzniká potřeba formátů dat, které

-Jsou čitelné stroji i člověku

-Jsou srozumitelné stroji i člověku, nebo-li obsahují nejen vlastní data, ale i informaci o významu těchto dat

-Syntaxe a způsob práce s těmito soubory je nějakým způsobem normalizovaná (W3C konsorcium - konsorcium starající se o standardy na internetu)

V našem kurzu se zaměříme na formáty souborů používaných v prostředí Internetu pro **čitelný přenos dat**, což v současnosti jsou především **XML soubory** (nebo jejich fragmenty) (XHTML stránka je XML soubor), **JSON soubory**.

Formát XML souboru se objevil a rychle prosadil v druhé polovině 90 let s rozvojem Internetu.

Formát JSON se prosazuje v posledním desetiletí především jako nativní formát JavaScriptu pro přenos dat mezi serverem a prohlížečem, především s rozvojem asynchronního přenosu dat (tj. přenosu na pozadí zobrazené stránky v prohlížeči).

Co to znamená asynchronní přenos dat?

Motivace - příklad XML dokumentu 1:

```
<?xml version="1.0" encoding="windows-1250"?>
<seznam_panovniku>
    <panovnik rod="přemyslovec">
        <jmeno>Přemysl Otakar II.</jmeno>
        <titul>král český</titul>
        <panoval>
            <od>1253</od>
            <do>1278</do>
        </panoval>
    </panovnik>
</seznam_panovniku>
```

Zobrazit v IE:

Motivace1.xml

Otázky:

Setkali jste s XML?

Kde?

Dokument obsahuje dvojí druh informace:

- Vlastní data dokumentu (červeně)
- Metainformace o struktuře a významu dat (černě)

Například řetězec „**Přemysl Otakar II.** “ je jméno panovníka, neboť je zařazeno ve struktuře pod elementy „**panovnik**“, „**jmeno**“

Motivace - příklad XML dokumentu 2 (XHTML):

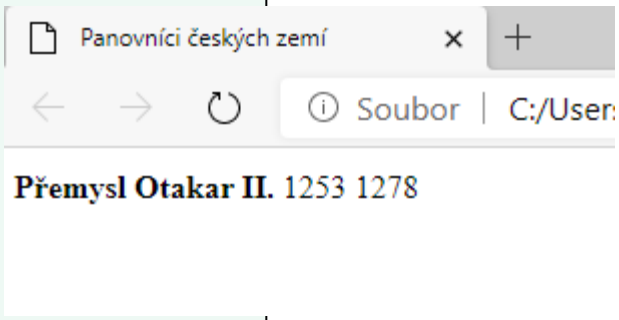
```
<HTML><HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Panovníci českých zemí</TITLE>
</HEAD>
<BODY>
    <table>
        <tr>
            <td><b>Přemysl Otakar II.</b></td>
            <td>1253</td>
            <td>1278</td>
        </tr>
    </table>
</BODY> </HTML>
```

Zobrazit v IE:

Motivace2.html

Otázky:

V čem se dva typy xml souborů liší?



Dokument obsahuje dvojí druh informace:

- Vlastní v prohlížeči viditelný obsah html stránky (červeně zobrazené údaje)

- Metainformace, které slouží k formátování těchto údajů do HTML tabulky (černě). Například řetězec „Přemysl Otakar II. “ je zobrazen v prvém sloupci a tučně

Motivace - zpracování:

```
<?xml version="1.0" encoding="windows-1250"?>
<seznam_panovniku>
  <panovnik rod="přemyslovec">
                <jmeno>Přemysl Otakar II.</jmeno>
                <titul>král český</titul>
                <panoval>
                        <od>1253</od>
                        <do>1278</do>
                </panoval>
        </panovnik>
</seznam_panovniku>
```

```
<HTML><HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Panovníci českých zemí</TITLE>
</HEAD>
<BODY>
        <table>
                <tr>
                        <td><b>Přemysl Otakar II.</b></td>
                        <td>1253</td>
                        <td>1278</td>
                </tr>
        </table>
</BODY> </HTML>
```

Setkali jsme se dvěma popisy stejných dat:

- Popis významu dat
- Popis formátování dat

Otázka: Potřebujeme to opravdu dvakrát?

Odpověď: Ne pokud umíme transformovat jeden popis do druhého.

Odpověď: XSLT transformace

Motivace - transformace:

```
<?xml version="1.0" encoding="windows-1250"?>
<seznam_panovniku>
  <panovnik rod="přemyslovec">
                <jmeno>Přemysl Otakar II.</jmeno>
                <titul>král český</titul>
                <panoval>
                        <od>1253</od>
                        <do>1278</do>
                </panoval>
        </panovnik>
</seznam_panovniku>
```

```
<HTML><HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Panovníci českých zemí</TITLE>
</HEAD>
<BODY>
        <table>
                <tr>
                        <td><b>Přemysl Otakar II.</b></td>
                        <td>1253</td>
                        <td>1278</td>
                </tr>
        </table>
</BODY> </HTML>
```

```
<xsl:template match="/">
   <HTML>
      <HEAD> <TITLE> Panovníci českých zemí    </TITLE>
</HEAD>
      <BODY>  <xsl:apply-templates/> </BODY>
   </HTML>
</xsl:template>

<xsl:template match=„seznam_panovniku">
<table border="border">
        <xsl:apply-templates/>
</table>
</xsl:template>

<xsl:template match="panovnik">
<tr>
<td><xsl:value-of select="jmeno"/></td>
<td><xsl:value-of select="panoval/od"/></td>
<td><xsl:value-of select="panoval/do"/></td>
</tr>
</xsl:template>
```

Co popisuje transformace:

- pokud jsem spuštěn vytvořím xhtml soubor
- pokud vstupní soubor obsahuje seznam panovníků, vytvořím tabulku
- pokud vstupní soubor obsahuje informace o panovníkovi, vytvořím řádek tabulky

Použita XSLT transformace.

Transformace je sama o sobě XML soubor.

Browser má zabudován nástroj, který umí transformaci provést.

Umí zobrazit xml soubor ve tvaru html, pokud připojíte konverzi požadovanou konverzi xslt

```
<xsl:template match="/">
   <HTML>
      <HEAD> <TITLE> Panovníci českých zemí    </TITLE>
</HEAD>
      <BODY>  <xsl:apply-templates/> </BODY>
   </HTML>
</xsl:template>

<xsl:template match=„seznam_panovniku">
<table border="border">
        <xsl:apply-templates/>
</table>
</xsl:template>

<xsl:template match="panovnik">
<tr>
<td><xsl:value-of select="jmeno"/></td>
<td><xsl:value-of select="panoval/od"/></td>
<td><xsl:value-of select="panoval/do"/></td>
</tr>
</xsl:template>
```

XML je značkovací jazyk (markup language).

V XML dokumentu se setkáváme s dvěma základními typy značek:

- s **elementy**, např. element<jmeno>. V našem dokumentu platí pravidlo, že text mezi počáteční a koncovou značkou <jmeno></jmeno> reprezentuje jméno panovníka.

<jmeno>Přemysl Otakar II.</jmeno>

- s **atributy**, např. atribut rod má hodnotu, jejíž význam je v našem případě panovnický rod.

<panovnik rod="přemyslovec">

V XML dokumentu se později setkáváme s dalšími syntaktickými prvky:

- procesní instrukce

- poznámky aj.

Vlastní XML dokument je tvořen:

- **nepovinnou deklarací**, která musí být první značkou dokumentu

<?xml version="1.0" encoding="windows-1250" stadalone="yes"?>

- **vlastním tělem dokumentu**, který je vždy tvořen jedním **kořenovým elementem**, v našem případě

<seznam_panovniku> …. </seznam_panovniku>

Deklarace:

<?xml version="1.0" encoding="windows-1250"?>

Deklarace je procesní instrukce určená ke zpracování XML souboru XML procesorem:

- atributem version říkáme, jaké verzi standardu dokument odpovídá (1.0, 1.1)
- atributem encoding říkáme, jak je dokument kódován, implicitně UTF8

Otázky:

Jaký je rozdíl mezi následujícími znakovými sadami (kódováním)?

- windows-1250
- windows-1252
- UTF8
- UTF

Co je to znaková sada (kódová tabulka)?

Kolik znaků lze kódovat pomocí 8 bitů?

Kolik znaků lze kódovat pomocí 16 bitů?

Které sady jsou 8bitové a které 16bitové?

XML procesorem zde rozumíme libovolný program zpracující XML data

Standard XML

Vzniká koncem 90 let (Standard XML 1.0 je z roku 1999, 1.1 z roku 2004)

Standard XML je odpověď na dobovou poptávku po univerzálním formátu dat pro sdílení a výměnu informací v otevřeném prostředí (Internet/Intranet).

Oproti uzavřeným (proprietárním) systémům minulosti (svět IBM, svět Microsoft, svět Apple) je internet otevřené prostředí s uvolněnými nebo mnohem uvolněnějšími vzájemnými vazbami.

Otevřený svět Internetu se stále nekoordinovaně mění a všichni hráči se musí neustále přizpůsobovat a současně vyvíjet obrovské úsilí se dohodnout (konsorcium W3C). Všichni jsou pod tlakem požadavku prostřednictvím internetu sdílet informace a služby, které jsou zpracovatelné programy (např. webové služby).

- XML je otevřený formát, který není spjat s žádnou platformou, výrobcem nebo konkrétní technologií.

- XML dokument si pomocí značek nese metainformace o obsahu dokumentu. Syntaxe a sémantika značek (včetně struktury dokumentu) je natolik bohatá, že umožňuje vytvářet „vlastní jazyky".

- XML standard umožňuje definici struktury pro třídu dokumentů (např. html dokument, open office dokument aj.)

-Od počátku je XML navržen pro mezinárodní použití, implicitním kódováním je Unicode (UTF8, 16 bitová mezinárodní znaková sada)

- XML dokument je zpracovatelný počítačem a do jisté míry složitosti a velikosti také čitelný člověkem.

XML jazyky (třídy dokumentů)

XML standard umožňuje definici struktury třídy dokumentů pomocí DTD (**D**okument **T**ype **D**efinition) a následnou validaci konkrétního dokumentu, tj. ověření zdali tento soubor patří do této třídy, tj. splňuje určité syntaktické podmínky na strukturu dokumentu.

Příklad1: XHTML soubor je soubor zpracovatelný a zobrazitelný webovým prohlížečem. Jeho elementy a atributy a jejich vzájemné vazby jsou omezeny, představují požadavky na kódování XHTML stránek.

Příklad2: Soubor popisující panovníky může být libovolný soubor splňující následující DTD strukturu:

```
<!ELEMENT seznam_panovniku (panovnik*)>
<!ELEMENT panovnik (jmeno+, titul, panoval?)>
<!ELEMENT panoval (od, do)>
<!ELEMENT jmeno (#PCDATA)>
<!ELEMENT titul (#PCDATA)>
<!ELEMENT od (#PCDATA)>
<!ELEMENT do (#PCDATA)>
<!ATTLIST panovnik rod CDATA #REQUIRED>
```

DTD je součást standardu XML.

Při výměně dat v otevřeném prostředí je třeba data na vstupu vždy validovat, a chránit tak svoje vnitřní prostředí. Formát DTD se neukázal příliš vhodný, dnes se více používá XML schémat.

Příklady jiných formátů používaných v kontextu DB:

CSV (Comma Separated Value).

Formát je vhodný na export jedné tabulky databáze. Setkáváme se s ním v oblasti DB nebo spreadsheets (excel).

Příklad ukazuje export tabulky zbozi pomocí pgAdmin, oddělovačem polí je středník:

```
"idzbozi";"popis";"cenakus"
5001;"zimní rukavice";490
5002;"závodní kolo";95000
5003;"kolečkové brusle";1000
5004;"vybaveni tělocvičky";110000
5005;"fotbalový míč";1050
5006;"volejbalový míč";1230
5007;"diabolka";10
5008;"stan";6800
5009;"pinpongový stul";12000
5010;"běžky Fischer";7500
```

Příklady jiných formátů používaných souběžně s XML:

JSON (JavaScript Object Notation).

Formát dat nezávislý na počítačové platformě určený pro přenos dat, která jsou organizována v polích nebo agregována v objektech.

Typické použití je JavaScript v prostředí internetu pro asynchronní přenos dat technologií AJAX. Jedná se o asynchronní technologii komunikace klienta (browser) a serveru (HTTP server).

Příklad:

```
{  "města": [
   {      "jméno": "Praha",      "populace": 1272690    },
   {      "jméno": "Brno",      "populace": 384277    }
]}
```

Očekávání od standard XML:

- Umožnit definovat třídy dokumentů (XML jazyky) a následně zajistit validaci těchto dokumentů různými technologiemi (DTD, XML Schémata, XDefinice, Relax NG)

- Poskytnout technologie pro práci s XML – Další vývoj naplnil toto očekávání prostřednictvím transformačních technologii (XSLT), možnosti adresace částí XML (XPath) a dotazovacích jazyků (XQuery). Příště se seznámíme s XPath a jeho použitím v SQL pro přístup na obsah xml položek v databázi.

- Informační obsah metadat (reprezentované XML značkami: elementy, atributy) – V XML dokumentu je možno pomocí značek definovat formátování (HTML) dokumentu nebo definovat význam vložených dat. Například náš příklad definuje třídu dokumentů, pomocí které lze popsat základní údaje o panovnících.

- Snadná zpracovatelnost strojem – V současné době většina jazyků, platforem (Java, JavaScript, …), SQL práci s XML podporují.

- Kombinace několika XML "jazyků" v jednom dokumentu – Zavedením jmenných prostorů lze kombinovat více značkových sad (jazyků) v jediném dokumentu.

Příklady značkovacích jazyků v syntaxi XML:

XHTML – HTML v syntaxi XML. Striktnější pravidla pro značkování, značky musí být ukončeny a nesmí se překrývat. Formátovací jazyk.

SOAP – jazyk pro výměnu zpráv nad HTTP protokolem (internet, intranet).

WSDL– jazyk pro popis webových služeb v syntaxi XML.

XSLT – jazyk pro transformaci XML do XML, textu. Vlastní XSLT je XML jazyk.

BPEL – jazyk pro orchestraci webových služeb, opět popsaný pomocí syntaxe XML.

OpenDokument (ODF) – OASIS Open Document Format for Office Applications. Otevřený formát založený na XML určený pro ukládání a výměnu dokumentů vytvářených kancelářskými aplikacemi.

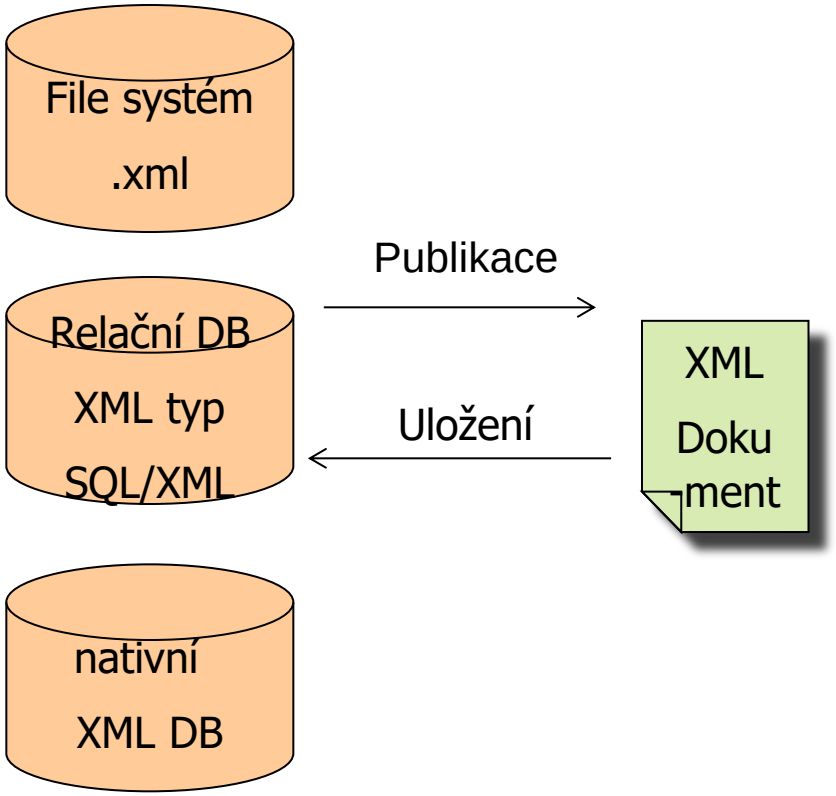
DocBook – Standard založený na XML určený především pro tvorbu dokumentace k hardware a software.

Xforms – XML formát pro vytváření webových formulářů.

MathML – Mathematical Markup Language, XML jazyk pro zápis matematických a podobných vzorců, W3C Projekt.

SVG – Scalable Vector Graphics. XML jazyk pro popis dvojrozměrné vektorové informace.

Perzistence XML dat



Zpracování XML dat

Adresace	XPath
Dotazování	XQuery
Transformace	XSLT
Validace	DTD XML Schema Relax NG Schematron
Parsing v programu	SAX, DOM


Thanks