Podpora XML v .NET

Jirka Kosek
nezávislý publicista
http://www.kosek.cz

Co nás čeká?

- podpora XML ve VisualStudio.NET
- architektura System.Xml
- čtení XML dokumentů
- generování XML dokumentů
- XSLT transformace
- práce s XML schématy

VS.NET jako XML editor

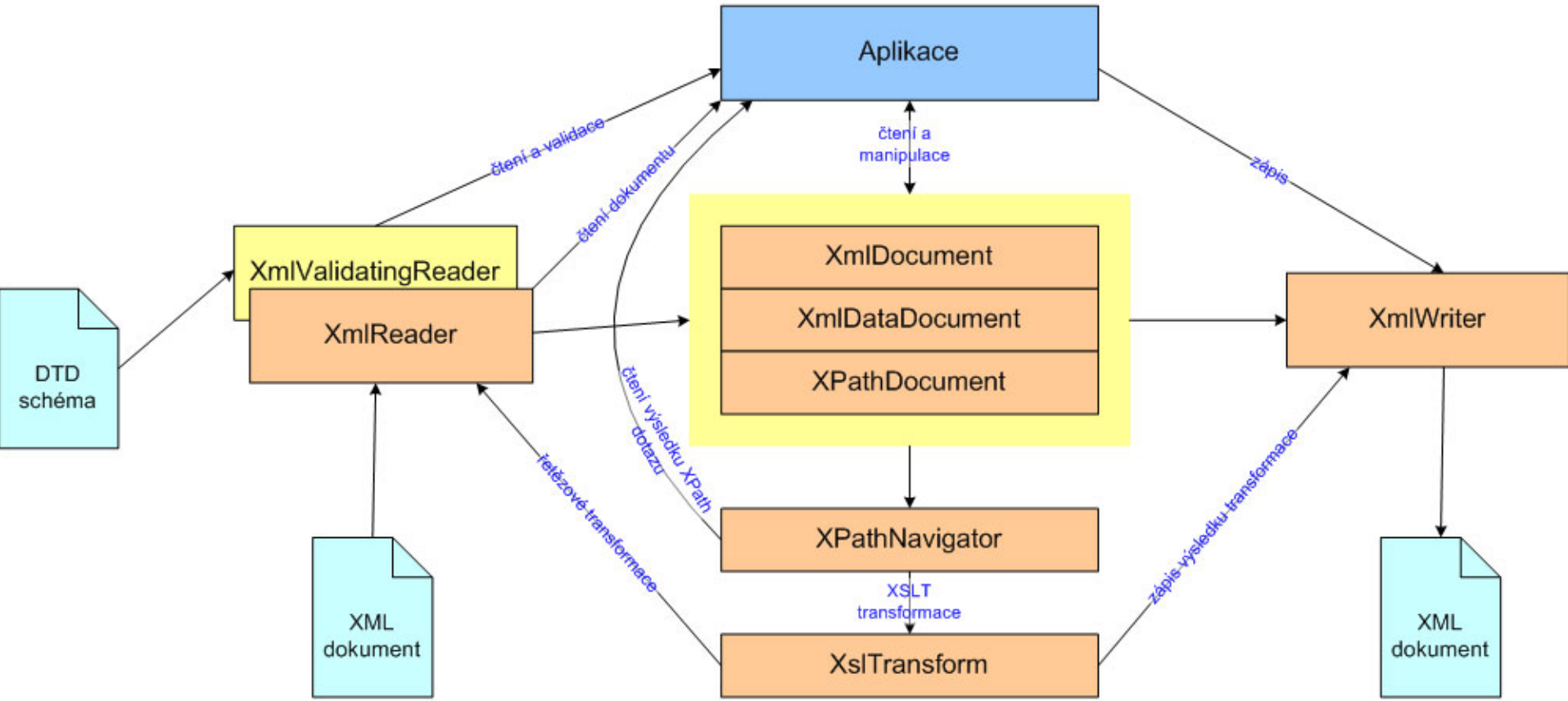
- téměř vše jsou nebo brzy budou XML dokumenty
  - konfigurační soubory
  - datové formáty pro výměnu dat (SOAP)
  - dokumentace
- vývojář potřebuje nástroj pro snadné editování XML dokumentů

VS.NET a XML

- demo
  - XML editor
    - textový a datový pohled
    - podpora XML schémat
    - validace
  - editor XML schémat
    - vizuální návrh schémat
    - import z databáze

Microsoft .net

System.Xml – architektura

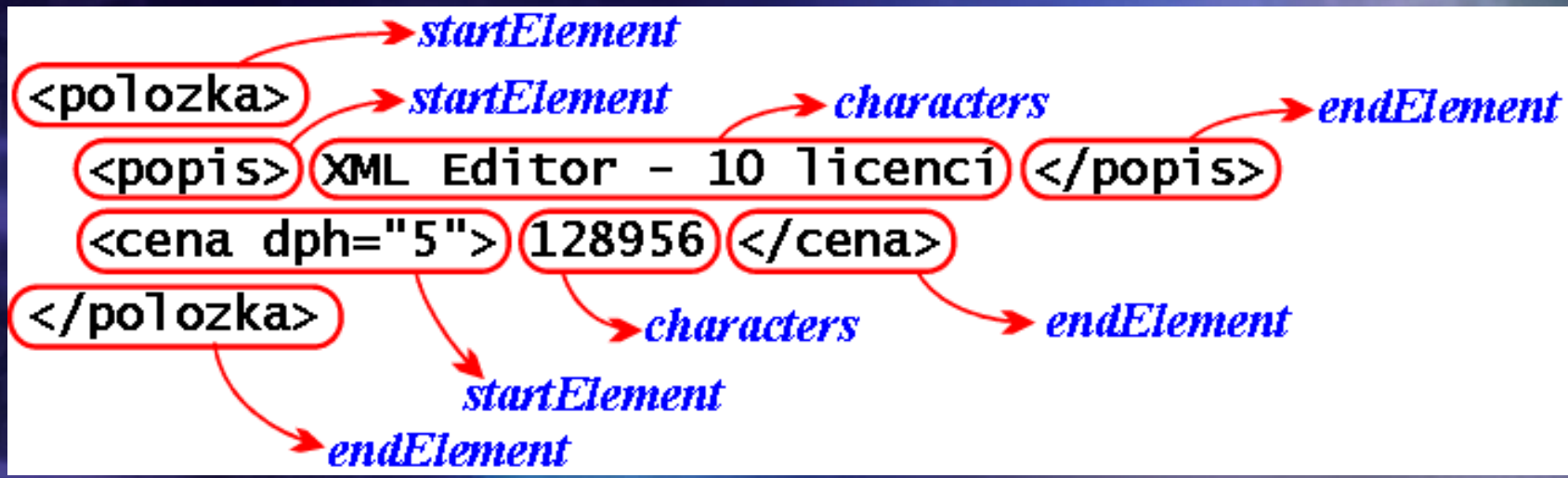


Čtení XML dokumentů

- parser
  - čte XML dokument a zpřístupní jej jako infoset pomocí API
- infoset
  - abstraktní model XML dokumentu
  - operuje na úrovni elementů, atributů, obsahu elementů, …

SAX – událostmi řízené čtení

- sekvenční čtení dokumentu
- rychlé a paměťově nenáročné
- jednotlivé části XML dokumentu vyvolávají události, které musíme obsloužit



Proč není SAX v System.Xml

- nevýhody
  - pro jednoduché dokumenty je psaní call-back funkcí zbytečně zdlouhavé
  - pro složité dokumenty jsou call-back funkce složité a musíme používat mnoho pomocných stavových proměnných
  - push parser
- v .NET si můžeme SAX implementovat sami jako obálku nad XmlReaderem

XmlReader

- moderní pull-parser
- části XML dokumentu čteme sekvenčně podle svých potřeb
- rychlý a paměťově nenáročný
- demo: sečtení faktury

Microsoft .net

XmlReader

- princip pull rozhraní:

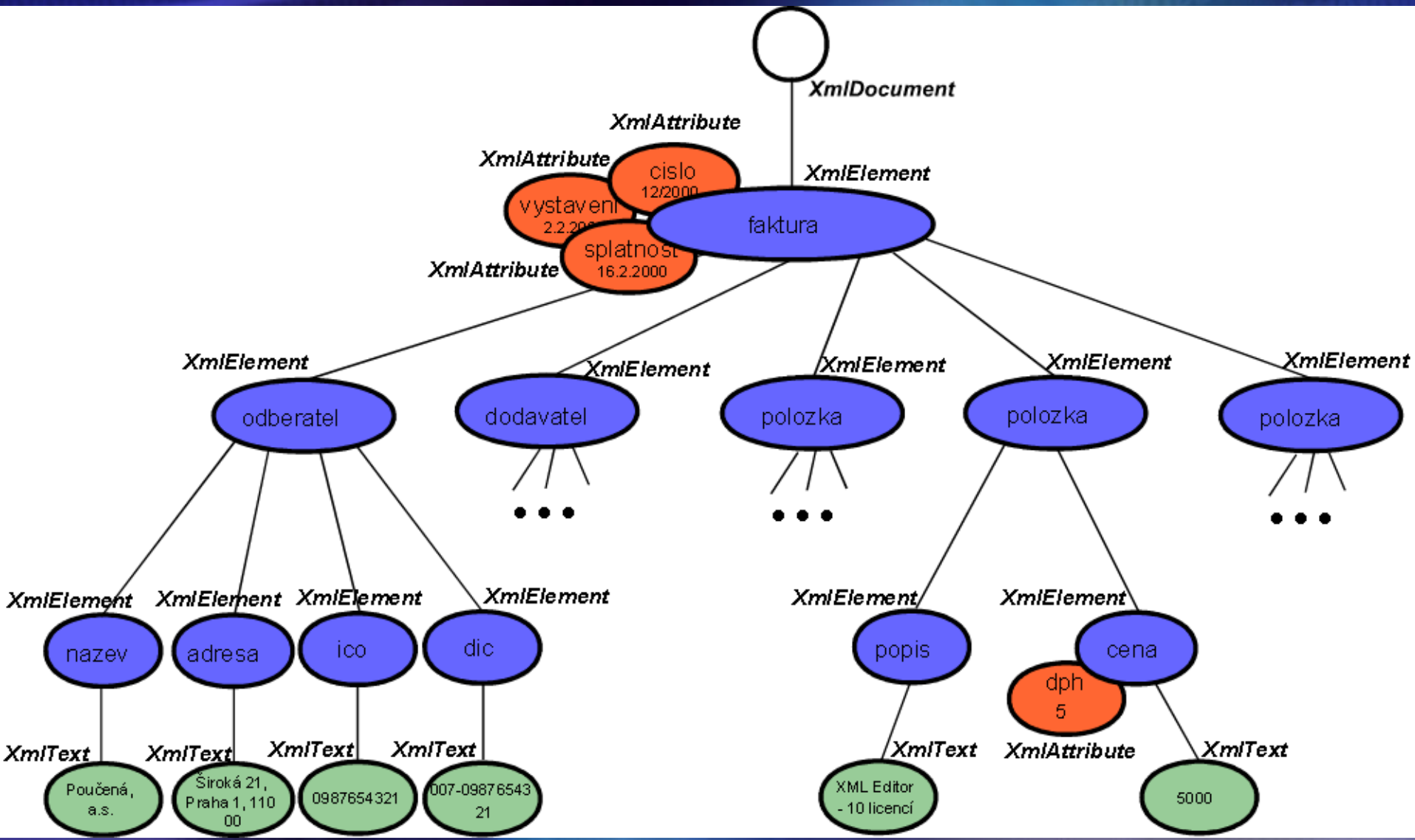
```
while (reader.Read())
{
    // zpracování uzlu
}
```

- nejdůležitější vlastnosti a metody
  - NodeType – typ uzlu
  - Name – jméno uzlu
  - ReadString() – přečtení obsahu el.
  - GetAttribute() – přečtení atributu

Rozhraní DOM

- celý dokument je zpřístupněn jako hierarchie objektů (XmlDocument, XmlNode, XmlElement, …)
- dokument můžeme opakovaně a nelineárně procházet
- dokument můžeme v paměti modifikovat
- velká paměťová náročnost, pomalejší než událostmi řízené
- pro chybný dokument se stromová reprezentace nevytvoří
- standard W3C

Ukázka DOM stromu



DOM – čtení dokumentu

- informace o uzlu
  - NodeType – typ uzlu
  - Name – jméno uzlu
  - Value – hodnota uzlu
- pohyb po stromu dokumentu
  - ParentNode
  - ChildNodes
  - FirstChild
  - LastChild
  - PreviousSibling
  - NextSibling
  - HasChildNodes

DOM – rozšíření MS

- načtení a uložení DOM stromu do souboru – metody Load(), Save()
- InnerText – textový obsah elementu včetně vnořených uzlů
- InnerXML – přístup k obsahu elementu jako k fragmentu XML
- demo: sečtení faktury pomocí DOM

DOM – modifikace dokumentu

- vytváření nových uzlů
  - CreateElement() CreateAttribute()
  - CreateTextNode() CreateCDataSection()
  - CreateComment() …
- modifikace stávajícího DOM stromu
  - AppendChild() ReplaceChild()
  - RemoveChild() CloneNode()
  - InsertBefore() InsertAfter()
- demo: vytvoření dokumentu v paměti

XmlDataDocument

- potomek XmlDocument
  - lze s ním manipulovat stejně jako s DOM stromem
  - lze na něj aplikovat XSLT transformace
- je dynamicky svázán s DataSetem
  - podle potřeby můžeme s relačními daty pracovat jako se záznamy nebo jako s XML dokumentem

XmlWriter

- pomocník při generování XML dokumentů
- ošetří generování deklarací jmenných prostorů
- snadné generování XML ze starších aplikací
- demo: zápis XML dokumentu do souboru

Microsoft .net

XSLT transformace

- kdy je potřeba transformovat XML dokumenty
  - při převodu zpráv mezi IS s jinou strukturou XML zpráv
  - při zobrazování XML – převod do HTML, XHTML, WML, FO, …
- XSLT
  - transformační jazyk
  - standard W3C
  - XML → XML, XML → HTML, XML → text

XslTransform

- velmi rychlá implementace XSLT
- vstupní XML
  - objekt XPathNavigator
    - obálka nad dalšími druhy XML objektů (např. nad DOM dokumentem)
- styl
  - XmlReader, URL, XPathNavigator
- výstup
  - XmlWriter, URL, XmlReader, TextWriter, Stream
- demo: převod faktury do HTML

XPathNavigator

- umožňuje vyhodnocování XPath výrazů nad XML dokumentem
- XPath
  - jednoduchý dotazovací jazyk
  - standard W3C
- Select() – výběr množiny uzlů
- Evaluate() – vyhodnocení výrazu
- demo: sečtení faktury

Microsoft .net

Validace XML dokumentů

- validace = ověření shody dokumentu se schématem
- podporované jazyky pro popis schématu
  - DTD, XDR, XSD (XML schémata)
- PSVI
  - Post Schema Validation Infoset
  - otypovaný infoset
  - vznikne po validaci infosetu oproti schématu

XmlValidatingReader

- validační vrstva nad XmlReaderem
- lze použít i pro DOM dokumenty
  - doc.Load(XmlValidatingReader)
- PSVI je dostupný přes metodu ReadTypedValue()
- chyba v dokumentu je událost
- schémata mohou být v cache
- demo: validace dokumentu

Schema Object Model

- SOM – objektový model pro práci s XML schématy
- čtení, modifikace a zápis schémat
- dostupné v System.Xml.Schema
- XmlSchemaCollection
  - cache pro XSD a XDR schémata
  - zrychlení aplikace při opakovaném použití stejného schématu

Podpora standardů W3C

- XmlReader, XmlDocument
  - XML 1.0 + Namespaces
  - DOM 2.0
- XPathNavigator
  - XPath 1.0
- XslTransform
  - XSLT 1.0
- XmlSchema
  - XML Schema 1.0 (XSD)

Modulární architektura

- System.Xml je založen na rozhraních a abstraktních třídách
  - pro části řetězce zpracovávajícího XML si můžeme napsat vlastní implementace a propojit je s ostatními komponentami
    - podpora katalogových souborů (XmlResolver)
    - vlastní potomek XmlReader pro čtení z jiných zdrojů dat než je XML
    - …