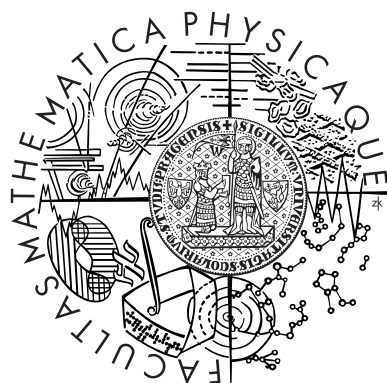


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tatsiana Maksimenka
Nástroj pro vizualizaci XQuery dotazů

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Irena Mlýnková, Ph.D.

Studijní program: Informatika

Obor: Obecná informatika

Praha 2012

Poděkování

Ráda bych poděkovala RNDr. Ireně Mlýnkové, Ph.D. za vedení mé práce, věnovaný čas a za podmětne připomínky. A také ráda bych poděkovala mé matce za podporu během mého studia.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 3. srpna 2012.

.....

Název práce: Nástroj pro vizualizaci XQuery dotazů

Autor: Tatsiana Maksimenka

Katedra: Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Irena Mlýnková, Ph.D., Katedra softwarového inženýrství

Abstrakt: Implementace nástroje pro podporu výuky jazyka XQuery s možnostmi vizualizace jak vstupních dat, tak i samotného dotazu a jeho výsledku.

Klíčová slova: XQuery, XML, vizualizace, graf

Title: An XQuery Visualization Tool

Author: Tatsiana Maksimenka

Department: Department of Software Engineering

Supervisor: RNDr. Irena Mlýnková, Ph.D., Department of Software Engineering

Abstract: Implementation of a tool to support teaching of XQuery language. Tool supports visualisation of input data, query itself and its results.

Keywords: XQuery, XML, visualisation, graph

Obsah

1	Úvod	7
1.1	Cíl práce	7
1.2	Použité technologie	8
1.3	Struktura textu	8
2	Existující nástroje	9
2.1	Altova XML Spy 2012	9
2.2	<oXygen/> XML Editor 14.0	9
2.3	Stylus Studio	11
2.4	XQueryViz	12
3	Implementace XQuery v .NET	15
3.1	Rozdíl mezi XPath 1.0 a XPath 2.0	15
3.2	Dokumenty XML v .NET	15
3.2.1	XmlDocument	16
3.2.2	XPathDocument	16
3.3	Saxon	16
3.4	QueryMachine	16
3.4.1	QueryMachine.XQuery	16
3.5	Srovnání knihoven pro .NET	17
4	XQueryTool	19
4.1	Z pohledu uživatele	19
4.1.1	Uživatelské rozhraní	19
4.1.2	Vizualizace XML dokumentu	20
4.1.3	Vizualizace XQuery dokumentu	25
4.1.4	Jak začít pracovat s XQueryTool	25
4.2	Architektura aplikace	27
4.3	XQuery v XmlGraphView	27
4.4	Další vývoj aplikace	28
5	Závěr	29
A	Obsah příloženého CD	30
	Literatura	31

Seznam obrázků

2.1.1	Hlavní okno Altova XML Spy 2012	10
2.2.1	Hlavní okno <oxygen> XML Editor	11
2.3.1	Hlavní okno Stylus Studio	13
2.4.1	Hlavní okno XQueryViz	14
4.1.1	XQueryTool: Hlavní okno	21
4.1.2	Zpracovávání XML dokument	22
4.1.3	XML graf	23
4.1.4	Vygenerovaný DOT soubor	23
4.1.5	Textová reprezentace XQuery dokumentu	24
4.1.6	sémantický graf XQuery	25
4.2.1	Class diagram klíčových částí aplikace XQueryTool	27

Kapitola 1

Úvod

XQuery¹ je výkonný a vhodný jazyk, určený pro zpracování XML² [16] [13] dokumentů. Pod pojmem XML dokumentu se rozumí nejenom soubory formátu XML, ale také databázi, jejich strukturu tvoří vnořené pojmenované stromy s atributy.

Jazyk XQuery (verze 1.0)[18] je standard definovaný konsorciem W3C [10], určený pro dotazování nad XML daty, je založený na XPath 1.0 [11] a XPath 2.0 [17]. XQuery je funkcionální jazyk, skládající se z několika druhů výrazů, které lze použít v různých kombinacích. Podporuje schémata, což je významným přínosem oproti ostatním dotazovacím jazykům, typový systém je založen na XML Schema [12]. XQuery je kompatibilní s ostatními dotazovacími jazyky nad XML daty.

V současné době existují softwarové produkty, podporující jazyk XQuery, většina z nich jsou komerční a jsou editory, které poskytují možnosti editování textové reprezentace XQuery dokumentů, spouštění dotazu, vizualizaci výsledku. Některé nástroje poskytují možnosti ladění dotazu, krokování a profilování³.

Z pohledu grafického rozhraní, je situace horší, skoro neexistují nástroje s podporou grafické vizualizace XQuery dotazu. Analýza vybraných softwaru je v kapitole 2.

1.1 Cíl práce

Cílem této práce je vyřešit dvě důležité úlohy: vizualizace textové a grafické reprezentace XQuery dokumentů a výsledku dotazu. Výsledkem XQuery dotazu je XML dokument. Na rozdíl od většiny grafických rozhraní, které vizualizují XML dokumenty a schémata ve tvaru adresářového stromu, vizualizace XML dokumentů v aplikaci XQueryTool je v přirozenějším tvaru, a to shora dolů. Uživatel může vytvářet nové dotazy, editovat otevřené dokumenty, spouštět a dostávat výsledek v různých reprezentacích. Uživatel může ukládat na disk výsledek v textové podobě

¹XQuery - An XML Query Language <http://www.w3.org/TR/xquery/>

²XML - Extensible Markup Language (XML) 1.0

³(Profilování - je nástroj pro vyhledávání míst v programu, které jsou vhodné pro optimalizaci <http://www.fit.vutbr.cz/~martinek/clang/profiling.html>)

nebo jej také může exportovat do různých formátů jako PDF⁴, SVG⁵, PNG⁶ a další. Pro lepší orientaci v dokumentech aplikace podporuje zvýraznění syntaxe.

1.2 Použité technologie

Aplikace XQueryTool je naprogramována v prostředí Microsoft Visual Studio 2010 Professional a .NET Framework 4⁷ v programovacím jazyku Visual C# 2010⁸. Při vývoji byly použity následující volně dostupné technologie:

- QueryMachine[20], viz3.4
- QueryMachine.XQuery [22] [20]
- Graph# [4]
- QuickGraph [7]
- Graphviz [5]
- AvalonEdit [2]
- WPF

Pro správu zdrojových kódů byl použit decentralizovaný verzovací systém Git, přesněji jeho grafická nadstavba Git Extensions⁹, která se dobře integruje do MS Visual Studia.

1.3 Struktura textu

Druhá kapitola se zabývá analýzou existujících nástrojů, podporující jazyk XQuery, z pohledu editace a vizualizace. Ve třetí kapitole budou představeny implementace XQuery v .NET a jejich srovnání. Další, čtvrtá, kapitola o aplikaci XQueryTool zahrnuje popis uživatelského rozhraní, architektury aplikace, popis nejzajímavějších částí, nalezení a řešení problémů při vývoji aplikace a také možné budoucí rozšíření XQueryTool.

⁴PDF - Portable Document Format

⁵SVG - Scalable Vector Graphics

⁶PNG - Portable Network Graphics

⁷<http://msdn.microsoft.com/en-US/vstudio/default.aspx>

⁸<http://msdn.microsoft.com/en-us/vstudio/hh388566.aspx>

⁹Git Extentions lze stáhnout na <http://code.google.com/p/gitextensions/>

Kapitola 2

Existující nástroje

2.1 Altova XML Spy 2012

Altova XML Spy 2012 [1] je komerční produkt, který zahrnuje nástroj pro editaci XML dokumentů a podporu pro vývoj XQuery dotazů v souladu se souborem specifikací a vestavěného XQuery engine.

XML Spy podporuje pouze textovou reprezentaci XQuery dokumentu (obr. 2.1.1), žádné jiné pohledy nejsou k dispozici. Editor je vybaven vestavěnou znalostí syntaxe XQuery. Pro editace dotazu poskytuje kontextově citlivé vstupní pomocníky [Enter Helpers], včetně seznamu proměnných, které jsou aktualizovány dynamicky během úprav dotazu. Pro přehlednost editor obsahuje inteligentní funkce, jako je obarvení syntaxe, automatické doplňování kódu a mnoho dalších utilit, které umožňují vytvářet a upravovat XQuery dokumenty rychle a snadno, například nástroje pro izolace chyb, ladění a optimalizaci XQuery dotazů, čímž usnadňuje práci programátorů.

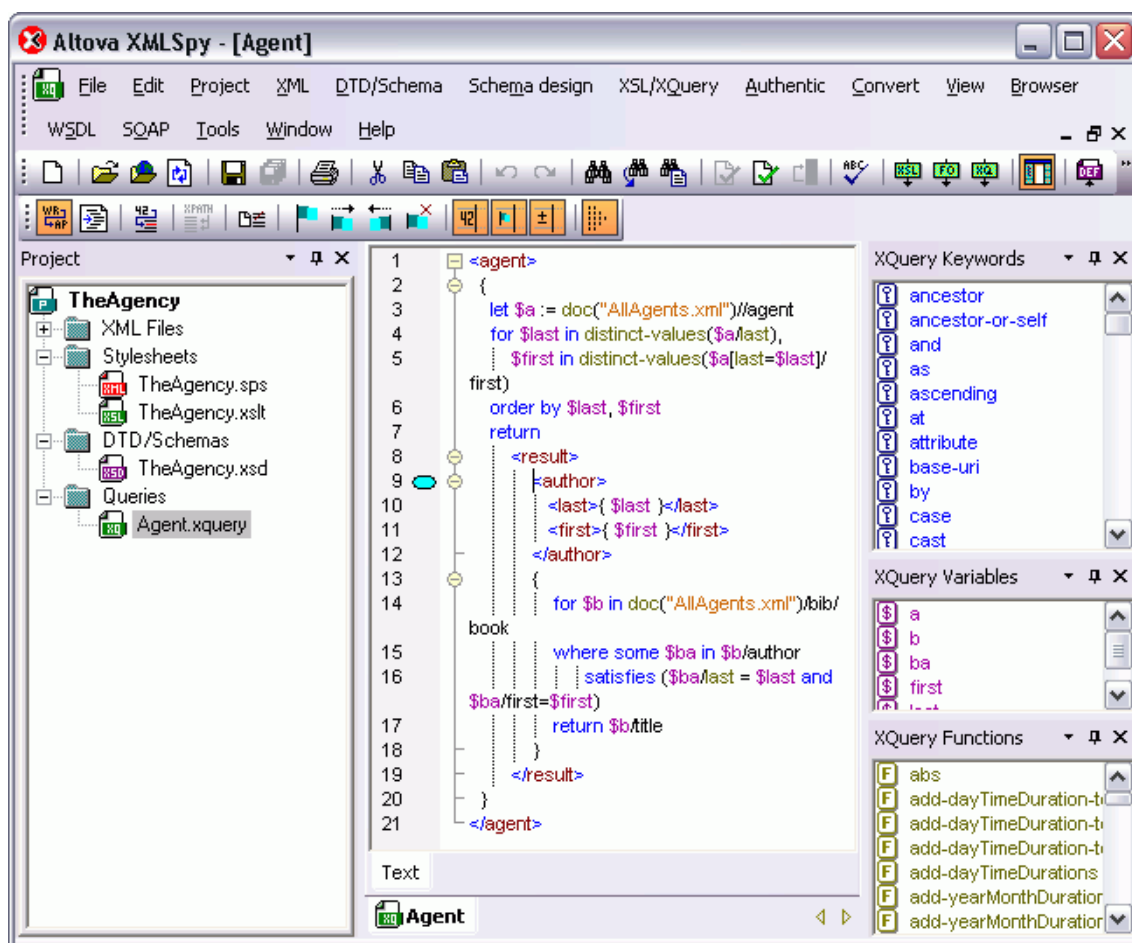
XQuery dotaz lze provádět nad aktivním XML dokumentem. Výsledkem dotazu je XML dokument, který je reprezentován standardními pohledy: text (Text View), mřížka (Grid View) a prohlížeč (Browser View). Textovou a mřížkovou reprezentaci lze upravovat, změny se objeví automaticky ve všech reprezentacích.

2.2 <oXygen/> XML Editor 14.0

XQuery je určen pro značení informací extrahovaných z různých zdrojů, jako strukturovaných XML dokumentů, objektů úložišť a relační databáze, včetně nativní XML databáze. <oXygen/> nabízí pomoc s úpravou, prováděním a laděním XQuery dokumentů.

<oXygen/> [6] poskytuje pouze textovou reprezentaci XQuery dokumentů, kterou lze upravovat. XML dokumenty lze editovat v XML editoru, který poskytuje následující reprezentaci XML dokumentů:

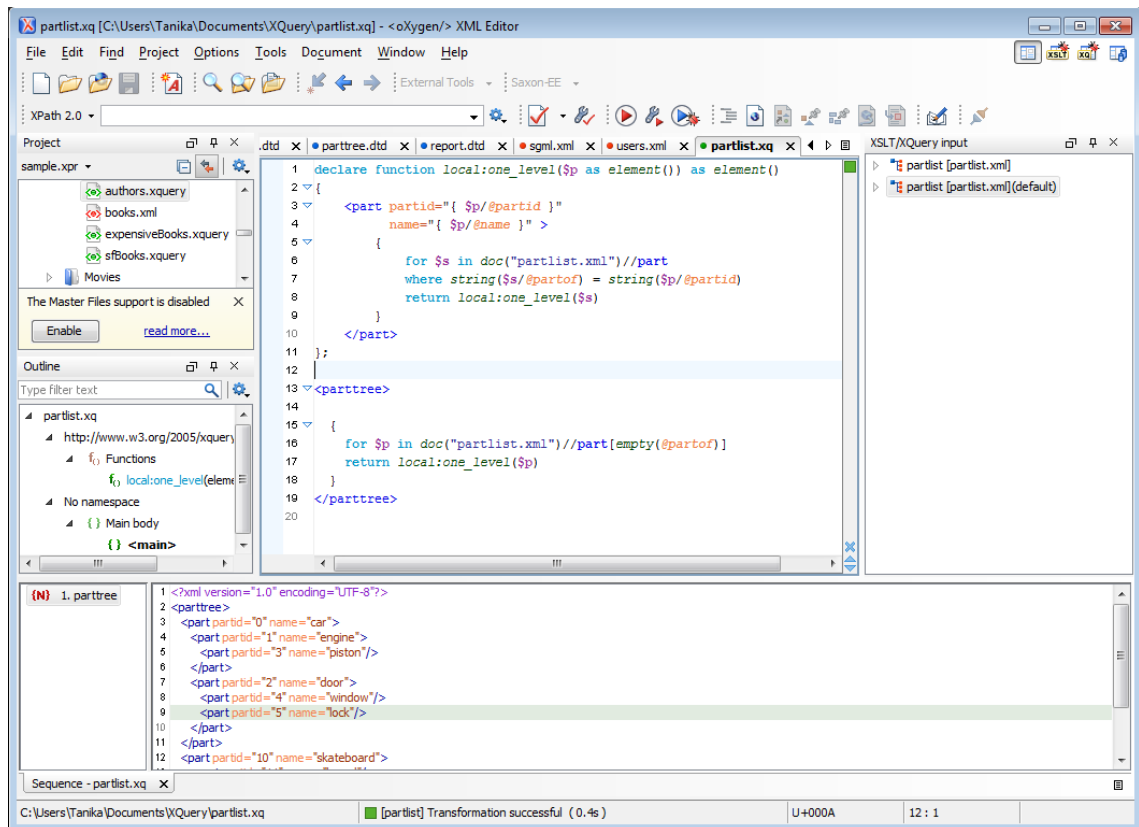
- text (Text View)
- mřížka (Grid View)



Obrázek 2.1.1: Hlavní okno Altova XML Spy 2012

- autor (Author)

Výsledkem dotazu je XML dokument, který na rozdíl od možnosti reprezentace XML dokumentu v XML editoru, lze zobrazit pouze v textové podobě obr. 2.2.1.



Obrázek 2.2.1: Hlavní okno <oxygen> XML Editor

2.3 Stylus Studio

Stylus Studio [9] poskytuje mnoho funkcí pro práci s XQuery (XML Query Language), včetně grafického mapovače, který umožňuje sestavit dotaz bez psaní jakéhokoli kódu a nástroje pro spouštění a ladění XQuery. Takže Stylus obsahuje nástroj profilování (XQuery Profiler), pro analýzu a optimalizaci XQuery dotazů, tím zvyšuje efektivitu zpracovávání dotazů.

V Stylus Studio pro práci s XQuery se používá textový editor a grafické rozhraní. Editor XQuery se skládá ze tří záložek: textová reprezentace XQuery dokumentů (XQuery Source), mapovač (Mapper) a plan (Plan).

Záložka zobrazující XQuery zdroj může být použita pro zobrazení, komponování, náhled a ladění XQuery.

Mapovač poskytuje rozhraní, které umožňuje vytvářet a graficky zobrazit XQuery dokument (obr. 2.3.1). A skládá se z:

- Panel zdrojových dokumentů, kde jeden nebo více zdrojových dokumentů může být přidáno.
- Panel cílové struktury, ve kterém můžeme určit strukturu požadovaného výsledku XQuery.
- Plátno mapovače, na kterém můžeme definovat podmínky, funkce a činnosti, pro uzly zdrojových dokumentů pro filtrování návratových hodnot, které jsou pak mapovány do cílového uzlu.
- Textový panel – umožňuje zobrazit kód XQuery při používání mapovače. Toto umožňuje sledovat jak změny v mapovači ovlivňují kód XQuery, aniž by bylo nutné přejít na záložku zdroje XQuery. Všechny zobrazení – Mapovače, Záložka zdroje XQuery a Textový panel – jsou synchronizovány.

Záložka Plan zobrazuje jak DataDirect XQuery procesor bude vykonávat XQuery kód a obsahuje informace o typech SQL které jsou používány pro přístup k relačním datům při XML streamingu, které dočasné tabulky vznikají když jsou volány proměnné, a tak dále.

Výsledek dotazu může být zobrazen kliknutím na tlačítko náhled výsledku (Preview Result). K dispozici jsou následující náhledy výsledku: stromová reprezentace (TreeView), náhled v prohlížeči, textová reprezentace. Také lze zobrazit profilování dat a uložit výsledek do souboru.

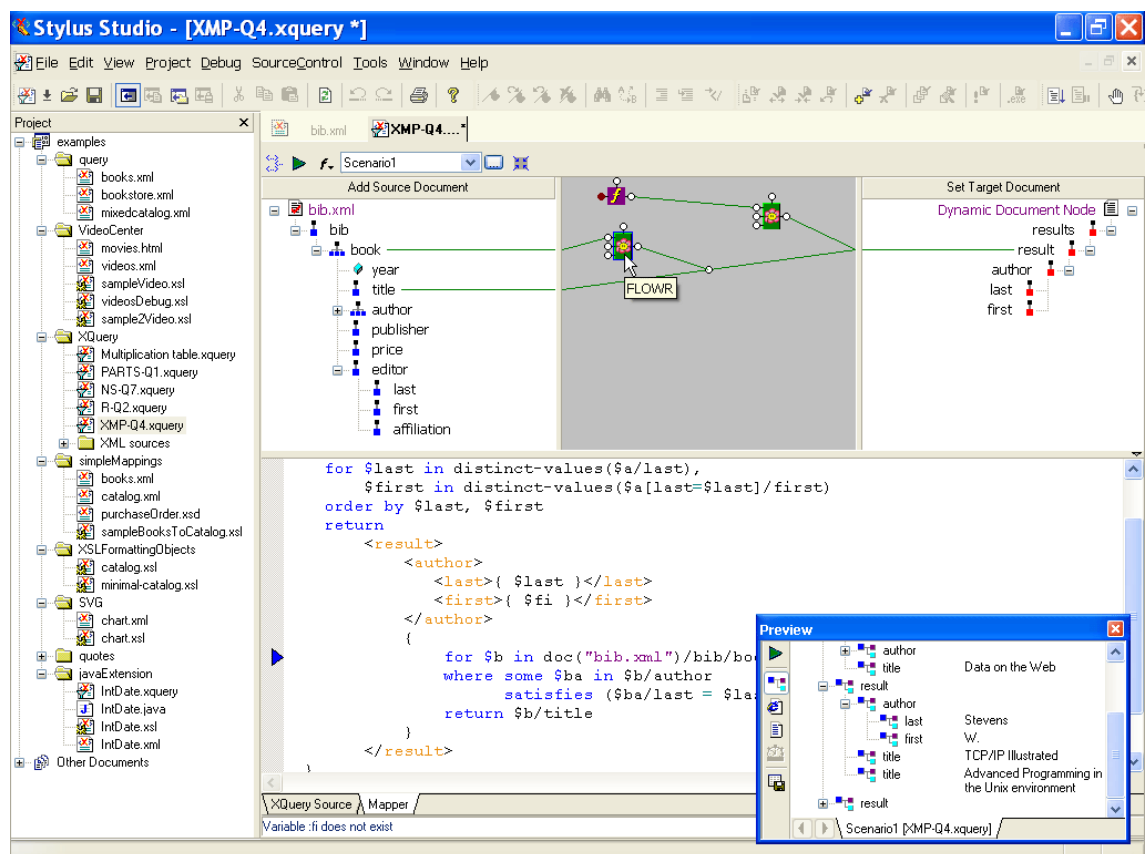
Stylus používá standardní prostředky pro vizualizace XML dokumentů, jsou to TextView, GridView, TreeView a SchemaView.

2.4 XQueryViz

XQueryViz [21] představuje grafický nástroj pro vizualizaci a konstrukci XQuery dotazů. Grafický nástroj může analyzovat jeden nebo více XML dokumentů/schémat, vizualizovat ve tvaru stromu, s možností přibližovat a oddalovat, expandovat nebo kontrahovat větve stromu. XQueryViz může také analyzovat textový XQuery dotaz a představit jej jako orientovaný acyklický graf.

Nástroj se skládá z následujících částí (obr. 2.4.1):

1. Grafická vizualizace XML dokumentů /schématu, vizualizace XPath dotazů – svislá pod-okna na levé straně.
2. Vizualizace klauzule „FOR-LET-WHERE“ (FLW) – svislá pod-okna uprostřed nástroje.
3. Vizualizace klauzule „RETURN“ (pod)dotazu XQuery – svislá pod-okna na pravé straně.
4. Textová reprezentace XQuery dotazu - vodorovné pod-okno.



Obrázek 2.3.1: Hlavní okno Stylus Studia

Vizualizace je přirozená, uživatel si může představit tok XML dat jako strom od kořene k listům, a tím usnadňuje pochopení a konstrukci složitějších XQuery dotazů.

Na rozdíl od XQBE, kde důraz je kladen na jednoduchost a cíloví uživatelé nejsou odborníci, XQueryViz je složitější a je určen pro pokročilejší uživatele.

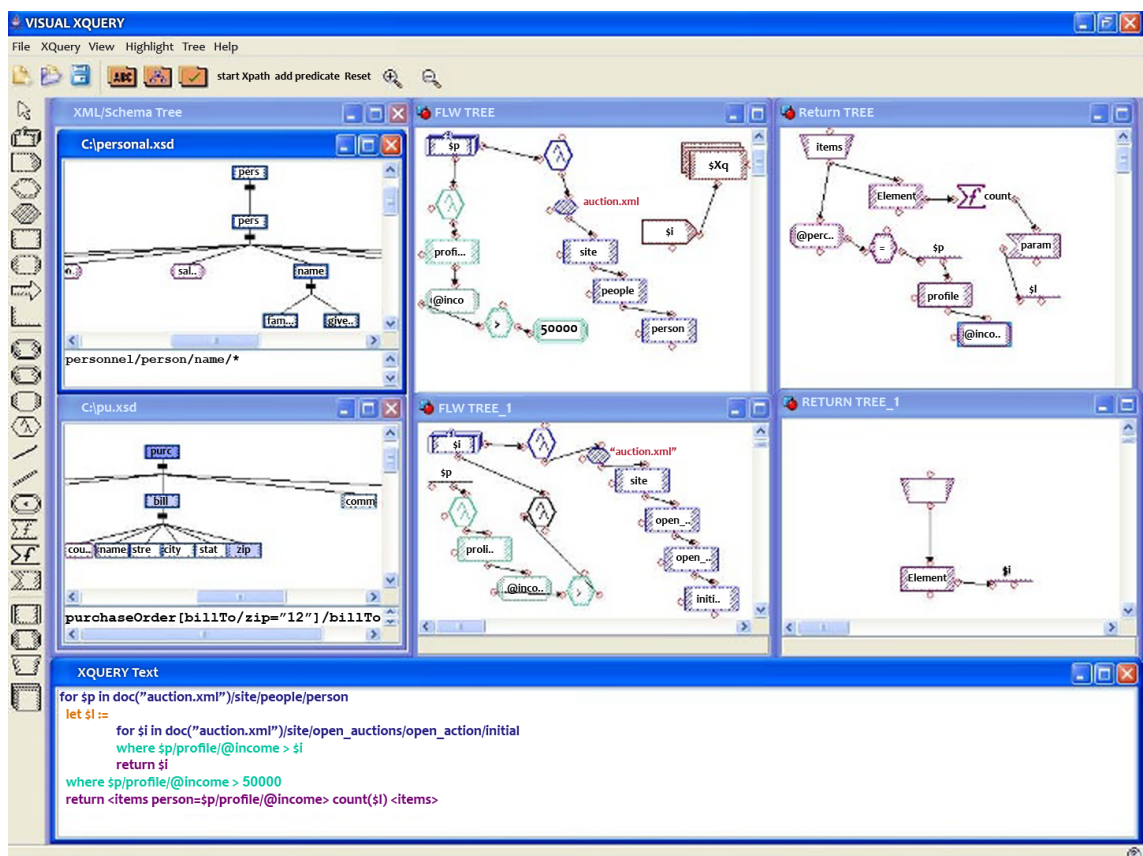
Uživatel může navedením a kliknutím vyrobit XPath dotaz, který se automaticky zobrazí v textové podobě pod stromem. Tento dotaz lze uložit a použít při konstrukci XQuery dotazu.

XQueryViz generuje z grafické reprezentaci dotazu textovou reprezentaci.

XQuery dotaz lze načíst ze souboru, vizuálně měnit, ukládat textový formát nebo zcela vybudovat nový dotaz s nebo bez použití nástroje konstrukce XPath. Konstruovat XQuery dotazy lze pomocí dílčích dotazů, které jsou konstruovány rekurzivně.

Nevyřešené problémy: Jak vizuálně propojit okna vizualizující dílčí konstrukci klauzuli „WHERE“ a „RETURN“, aby uživateli bylo jasno, který dílčí dotaz je součástí klauzule „WHERE“ nebo „RETURN“.

Tento grafický nástroj není k nalezení na internetu, existuje jenom článek o něm [21], takže program nebylo možné prozkoumat.



Obrázek 2.4.1: Hlavní okno XQueryViz

Kapitola 3

Implementace XQuery v .NET

Bohužel, v současné době, platforma .NET nemá nativní podporu XQuery. Architekti .NET jsou přesvědčení, že XQuery lze nahradit technologií LINQ-to-XML¹. Více o tom lze nalézt v blogu Microsoft XML Team's WebLog². XML třídy .NET však nepodporují XPath 2.03.1, který je podmnožinou XQuery, takže LINQ-to-XML nemůže nahradit XQuery, alespoň protože funkce rozšíření XPathSelectElement a XPathSelectElements jako parametry dostávají výrazy XPath 1.0.

3.1 Rozdíl mezi XPath 1.0 a XPath 2.0

XPath 1.0 pracuje s abstraktním datovým modelem XML dokumentu, který je chápán jako strom s uzly typů: dokument, element, atribut a dalšími. Na rozdíl od XPath 1.0 v XPath 2.0 datový model podporuje XML schéma. Navigace v dokumentu je pomocí cest. Výsledkem vyhodnocení výrazu nad stromem XML dokumentu je:

- V XPath 1.0 množina uzlů nebo hodnota typu řetězec, číslo nebo logická hodnota.
- V XPath 2.0 kolekce nebo hodnota typu XML schéma.

Navíc v XPath 2.0 se zavádí koncepce uspořádání uzlů v dokumentu, proto jsou definovány operátory „<<“ , „>>“ a „is“, které umožňují porovnávat umístění v pořadí původního dokumentu. Přidány jsou operátory intersect a except, relační operátory (eq, ne, gt, lt, atd.) a aritmetický operátor idiv. Rozšířen je i seznam vestavěných funkcí [19]. Další rozdíly lze najít v [17].

3.2 Dokumenty XML v .NET

Microsoft .NET Framework [je citace z XML efektivní programování pro .NET Dino Esposito [23]], kromě readerů a writerů XML, obsahuje také třídy, které analyzují

¹<http://msdn.microsoft.com/en-us/library/bb387098.aspx>

²Blog Microsoft XML Team's WebLog <http://blogs.msdn.com/b/xmlteam/archive/2007/02/08/standalone-xquery-implementation-in-net.aspx>

XML dokumenty podle objektového modelu W3C Document Object Model (DOM) [3]. Tyto třídy jsou dostupné ve jmenném prostoru System.XML.

3.2.1 XmlDocument

XmlDocument je nejčastěji používaná implementace objektového modelu, která vytvoří v paměti počítače stromovou reprezentaci dokumentu. Obsahuje metody pro načítání, ukládání XML dokumentu, validaci, manipulaci s uzly stromu a spustit XPath výrazy na obsah dokumentu. Výhoda použití této třídy spočívá v jednoduché manipulaci s uzly ve stromu. Hlavní nevýhodou je, že je to paměťově náročné.

3.2.2 XPathDocument

Třída XPathDocument zajišťuje vyšší výkon na rozdíl od XmlDocumentu, protože poskytuje optimalizovanou mezipaměť pro XML dokumenty a pracuje v režimu read-only.

3.3 Saxon

Saxon-HE (home edition) [8] je open source, který vyvinul Michael Kay pro Saxonica Limited. Poskytuje implementace XSLT 2.0 [15], XQuery 1.0 a XPath 2.0 pouze na základní úrovni definované W3C. Saxon-HE nenabízí funkce, které zahrnují schéma-aware.

Saxon je napsány v Javě a pro práci v .NET vyžaduje IKVM.NET³. V souvislosti s tím celková velikost assemblies potřebných ke spuštění Saxonu přesahuje 30 MB. Kromě režie související s potřebou využití dalších komponent, dělá tento přístup problematickým při použití standardizovaného XML dokumentu, což vede ke snížení výkonnosti a spotřebě paměti.

3.4 QueryMachine

QueryMachine⁴ je knihovna, která je „univerzálním“ ovladačem ADO .NET a určena pro provádění SQL dotazů výběru z jakéhokoli zdroje strukturovaných dat. Projekt je napsán v jazyku C# a používá společné platformy vypracované autorem XQuery procesoru [22], která je open source. Tyto nástroje jsou umístěny na serveru CodePlex⁵.

3.4.1 QueryMachine.XQuery

QueryMachine.XQuery [20] je open source, poskytuje implementaci XQuery 1.0 založené na třídě XPathNavigator API.

³IKVM.NET <http://www.ikvm.net/>

⁴QueryMachine documentation <http://qm.codeplex.com/documentation>

⁵CodePlex <http://qm.codeplex.com/>

Hlavní rysy implementace XQuery (citováno z [20]):

- založena na standardní třídě XPathNavigator API
- schéma-aware XQuery
- analýza dokumentu na požádání
- obrovské vstupní soubory nejsou zcela načteny do paměti
- XQuery výrazy jsou kompilované přímo do MSIL
- nativní podpora pro soubory MS Office OpenXML
- hash spojení FLWOR optimalizace a implicitní paralelizace XQuery požadavků
- podpora XML mapování (pouze experimentální)
- plně implementuje minimální plnění funkcí díky standardu W3C XML Query Language XQuery Test Suite výsledky na W3C⁶.

3.5 Srovnání knihoven pro .NET

Během vývoje XQueryTool bylo vyzkoušeno několik knihoven pro práci s XML a XQuery. Pro XQuery byly vyzkoušeny dvě knihovny Saxon a QueryMachine.XQuery, pro XML pak třídy XmlDocument (DOM), XPathDocument a XQueryDocument, o kterých bylo zmíněno v předchozích částech. Efektivita provádění dotazu nad XML daty, záleží na efektivitě zpracovávání XML dat, proto je zajímavé jejich srovnání, viz tabulka 3.2, vstupní XML dokumenty jsou popsány v tabulce 3.1.

Název souboru	input.xml	input2.xml	auctions.xml	book.xml
Velikost souboru	220 MB	11 MB	1 MB	1,4KB
Počet elementů count(doc(„Název souboru“)//*)	12 055 489	152 128	17 132	37
Počet uzlů XmlNode v tabulce dokumentu	33 559 875	306 091	46 269	98

Tabulka 3.1: Popis XML souborů

Z tabulky 3.2 je vidět, že soubor XQueryDocument na velkém souboru spotřebuje osmkrát méně paměti než XmlDocument a také načítá XML rychleji. Ve srovnání s XPathDocument nahrávání je o 25 % pomalejší a vyžaduje čtyřikrát méně paměti.

Ale pro malé soubory jsou lepší standardní reprezentace XML dokumentu. XQueryDocument se vydává, protože při načítání provádí více dalších akcí. Ale tento faktor však není kritický, protože doba nahrávání, ve většině případů je zanedbatelná na rozdíl od celkové doby provádění dotazu.

⁶<http://dev.w3.org/2006/xquery-test-suite/PublicPagesStagingArea/XQTSReportSimple.html>

Třída(x64,Intel Q9550,8GB RAM)	input.xml	doba nahrávání (n=1)	input2.xml	doba nahrávání (n=5)
XmlDocument	1.3 GB	12430 ms	91 MB	587 ms
XPathDocument	700 MB	8781 ms	61 MB	383 ms
XQueryDocument	150 MB	11739 ms	35 MB	592 ms
	auctions.xml	doba nahrávání (n=10)	book.xml	doba nahrávání (n=1000)
XmlDocument	20 MB	24,09 ms	-	0,35 ms
XPathDocument	20 MB	23,12 ms	-	0,34 ms
XQueryDocument	23 MB	59 ms	-	0,81 ms

Tabulka 3.2: Srovnání tříd pro práci s XML dokumenty

Kapitola 4

XQueryTool

XQueryTool je aplikace, která poskytuje možnosti práce s XQuery a XML dokumenty. XQuery dokumenty lze editovat a vizualizovat. XML dokumenty lze pouze vizualizovat.

4.1 Z pohledu uživatele

Po spuštění aplikace se uživateli otevře hlavní okno XQueryTool, viz obr.4.1.1.

4.1.1 Uživatelské rozhraní

V horní části hlavního okna je **Menu** (fialově zvýrazněno v obr. 4.1.1), obsahující následující funkce aplikace.

Nabídka menu **File** obsahuje položky:

- *New XQuery (Ctrl+N)* – vytvoří nový prázdný XQuery dokument v levé části okna.
- *Open (Ctrl+O)* – otevře dialogové okno pro otevření souboru, lze otevřít XML nebo XQuery soubor.
- *Save (Ctrl+S)* – uloží změny v souboru, ale v případě změn nově vyrobeného souboru XQuery zobrazí dialogové okno pro výběr nebo vytvoření nového názvu souboru.
- *Save As* – zobrazí dialogové okno pro výběr typu a názvu souboru.
- *Preferences* – zobrazí okno pro nastavení umístění XQuery souborů, a nastavení cest XML souborů, použitých v dotazu XQuery.
- *Exit* – ukončí program a zavře všechna okna.

Nabídka menu **Edit** obsahuje položky:

- *Undo* – zruší naposledy vykonanou akci.

- *Redo* – provádí návrat do původního stavu, po nějakém kroku pomocí Undo.
- *Cut, Copy, Paste* – operace se tykají textu v aktivním editoru: vyjmi, zkopíruj, vlož.
- *Select All* – označí celý text v aktivním editoru.

Nabídka menu **XQuery** obsahuje položky:

- *Execute (F5)* – spustí XQuery dotaz
- *Cancel (Esc)* – zruší provádění XQuery dotazu.

Nabídka menu **Help** obsahuje položky:

- *XQuery on w3c.org* – příručka jazyka XQuery
- *About* – informace o programu

Po levé straně si uživatel může nahrát adresář souborů (modře zvýrazněno v obr.4.1.1), pro práci s XQuery a XML soubory.

XML soubory se zobrazí do vedlejšího okna (červeně zvýrazněno v obr. 4.1.1), kde si uživatel může prohlédnout vizualizace XML dokumentů.

Po pravé straně je okno pro vizualizaci XQuery dokumentu (zeleně zvýrazněno v obr.4.1.1) a hned pod ním je okno výsledku dotazu (oranžově zvýrazněno v obr. 4.1.1).

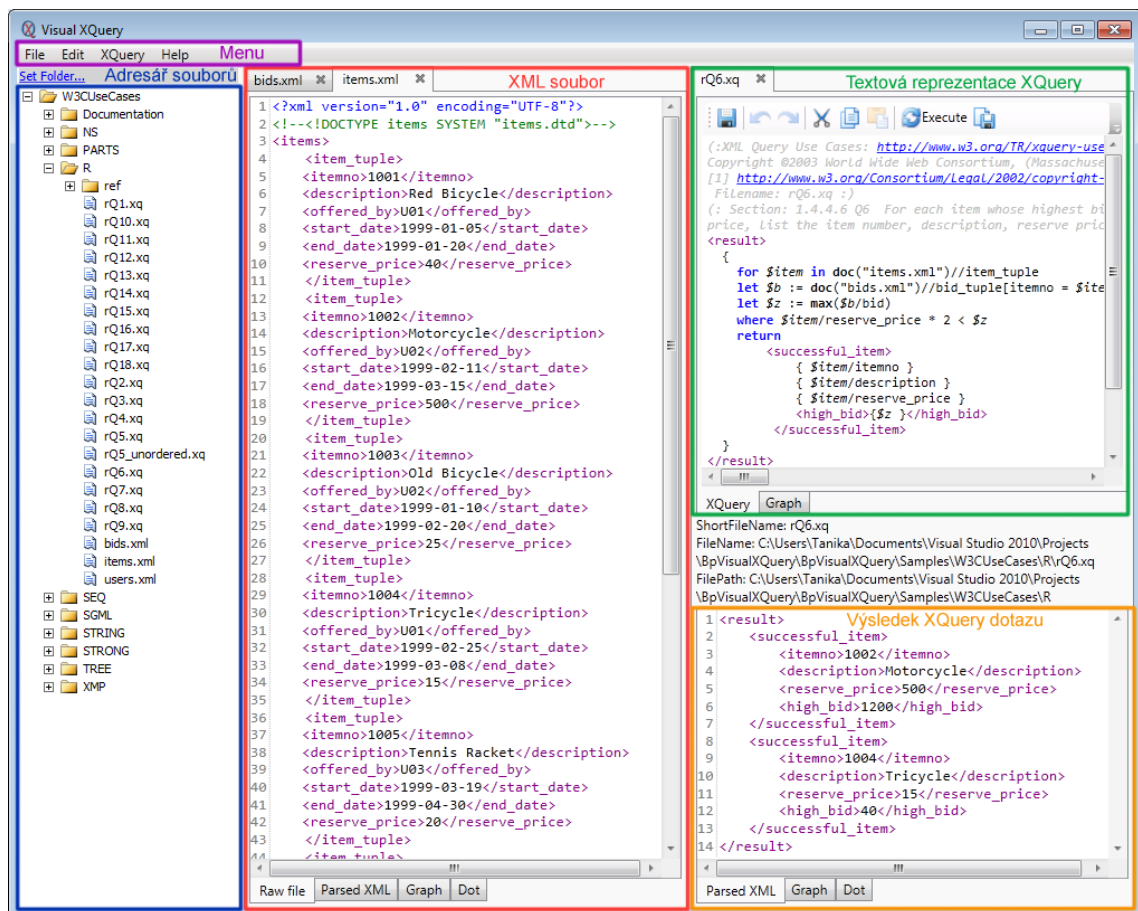
4.1.2 Vizualizace XML dokumentu

Při otevření XML dokumentu, program vizualizuje jej jako:

- *Raw file* - nezpracovávány XML soubor
- *Parsed XML* obr.4.1.2 – zpracovávány XML soubor
- *Graph* obr.4.1.3 – grafické představení XML souboru

Jednotlivé uzly XML dokumentu jsou v grafu znázorněny takto:

- *Dot* obr.4.1.4 – vygenerovány DOT soubor, pro exportování XML dokumentu do formátů PDF, SVG, PNG, JPEG atd.



Obrázek 4.1.1: XQueryTool: Hlavní okno


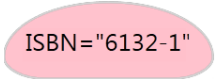
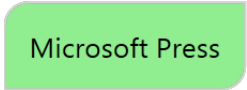
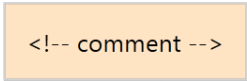
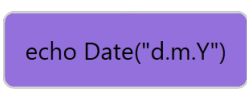
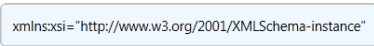
```

1 <result>
2   <successful_item>
3     <itemno>1002</itemno>
4     <description>Motorcycle</description>
5     <reserve_price>500</reserve_price>
6     <high_bid>1200</high_bid>
7   </successful_item>
8   <successful_item>
9     <itemno>1004</itemno>
10    <description>Tricycle</description>
11    <reserve_price>15</reserve_price>
12    <high_bid>40</high_bid>
13  </successful_item>
14 </result>

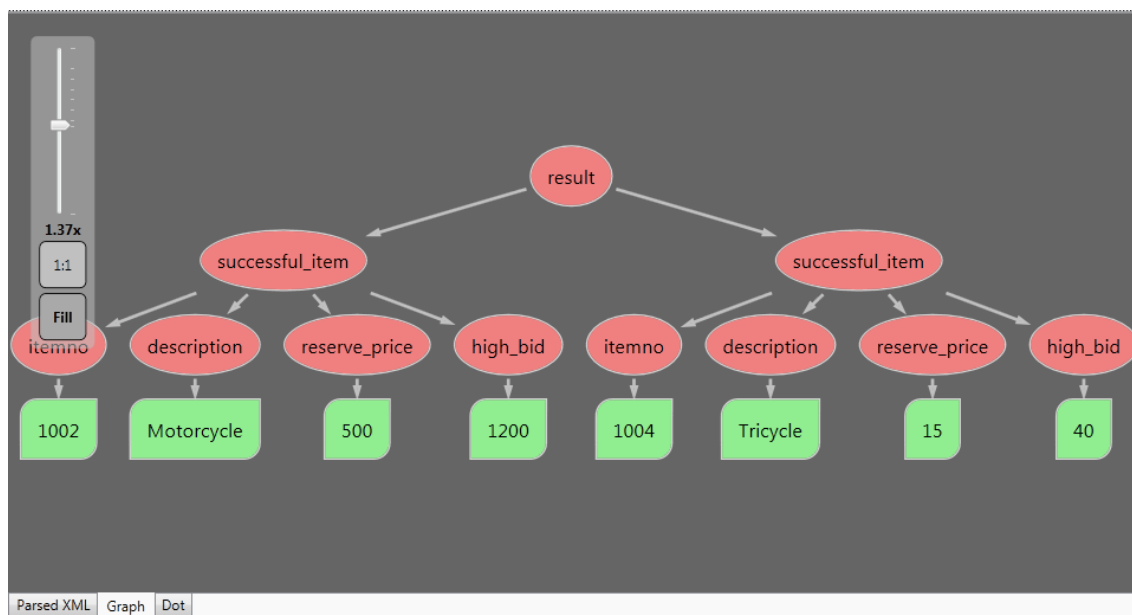
```

Parsed XML | Graph | Dot

Obrázek 4.1.2: Zpracovávány XML dokument

Typ uzlu	Vizualizace uzlu	Popis
Element		Uzel typu Element s názvem „publisher“.
Attribute		Uzel typu Attribute s názvem „ISBN“ a hodnotou „6132-1“.
Text		Uzel typu Text s hodnotou „Microsoft Press“.
Comment		Uzel typu Comment s hodnotou „comment“.
Processing Instruction		Uzel typu Processing Instruction obsahující příkaz „echo Date(„d.m.Y“)“.
Namespace		Uzel typu Namespace s prefixem „xmlns“, názvem „xsi“ a hodnotou „http://www.w3.org/...“.

Tabulka 4.1: Grafická reprezentace uzlů XML dokumentu



Obrázek 4.1.3: XML graf

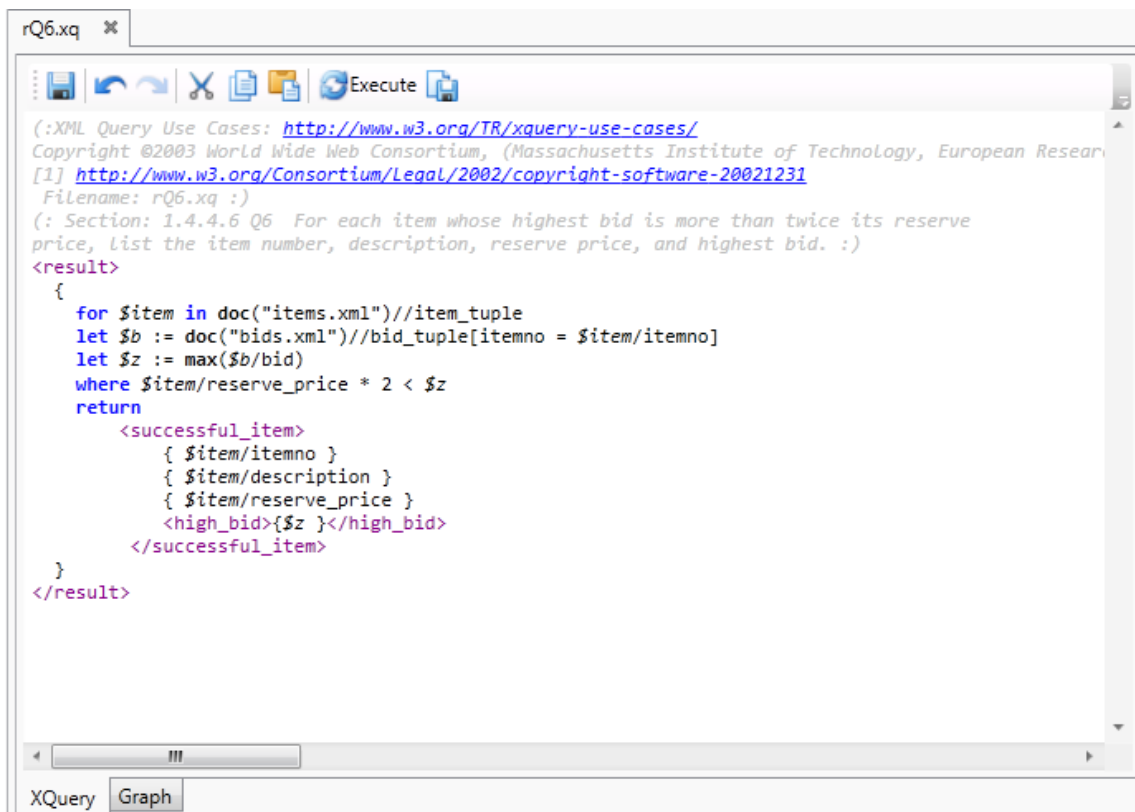
```

1 |digraph G {
2 |   ordering=out;
3 |   node[shape=egg,style=filled,color=black,fillcolor=white];
4 |   1[label="result",shape=egg ,fillcolor=coral];
5 |   2[label="successful_item",shape=egg ,fillcolor=coral];
6 |   1:_:s->2:_:n;
7 |   3[label="itemno",shape=egg ,fillcolor=coral];
8 |   2:_:s->3:_:n;
9 |   4[label="1002",shape=rect,fillcolor=lightgrey];
10 |  3:_:s->4:_:n;
11 |  5[label="description",shape=egg ,fillcolor=coral];
12 |  2:_:s->5:_:n;
13 |  3:_:e--5:_:w[style=invisible];
14 |  6[label="Motorcycle",shape=rect,fillcolor=lightgrey];
15 |  5:_:s->6:_:n;
16 |  7[label="reserve_price",shape=egg ,fillcolor=coral];
17 |  2:_:s->7:_:n;
18 |  5:_:e--7:_:w[style=invisible];
19 |  8[label="500",shape=rect,fillcolor=lightgrey];
20 |  7:_:s->8:_:n;
21 |  9[label="high_bid",shape=egg ,fillcolor=coral];
22 |  2:_:s->9:_:n;
23 |  7:_:e--9:_:w[style=invisible];
24 |  10[label="1200",shape=rect,fillcolor=lightgrey];
25 |  9:_:s->10:_:n;
26 |  11[label="successful_item",shape=egg ,fillcolor=coral];
27 |  1:_:s->11:_:n;
28 |  2:_:e--11:_:w[style=invisible];
29 |  12[label="itemno",shape=egg ,fillcolor=coral];
30 |  11:_:s->12:_:n;
31 |  13[label="1004",shape=rect,fillcolor=lightgrey];
32 |  12:_:s->13:_:n;
33 |  14[label="description",shape=egg ,fillcolor=coral];
34 |  11:_:s->14:_:n;

```

dot.exe -v -T png -o C:\export.png

Obrázek 4.1.4: Vygenerovaný DOT soubor



```
(:XML Query Use Cases: http://www.w3.org/TR/xquery-use-cases/
Copyright ©2003 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research
[1] http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231
Filename: rQ6.xq :)
(: Section: 1.4.4.6 Q6 For each item whose highest bid is more than twice its reserve
price, list the item number, description, reserve price, and highest bid. :)
<result>
{
  for $item in doc("items.xml")//item_tuple
  let $b := doc("bids.xml")//bid_tuple[itemno = $item/itemno]
  let $z := max($b/bid)
  where $item/reserve_price * 2 < $z
  return
    <successful_item>
      { $item/itemno }
      { $item/description }
      { $item/reserve_price }
      <high_bid>{$z}</high_bid>
    </successful_item>
}
</result>
```

XQuery Graph

Obrázek 4.1.5: Textová reprezentace XQuery dokumentu

4.1.3 Vizualizace XQuery dokumentu

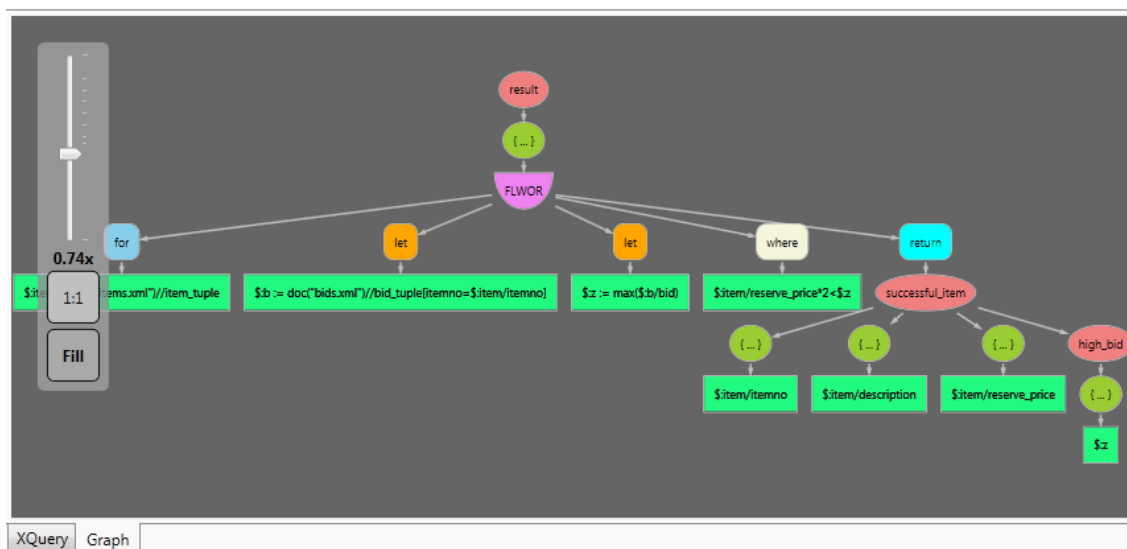
- *XQuery* obr.4.1.5 – textová reprezentace XQuery dokumentu nebo textový editor XQuery dokumentu.

XQuery editor obsahuje pomocné funkce, pro editaci textu: Save, Undo, Redo, Cut, Copy, Paste, Execute – mají stejný význam jako funkce v Menu Edit a Menu XQuery, navíc je definována funkce na uložení výsledku XQuery dotazu.

- *Graph* obr.4.1.6 – grafická reprezentace XQuery dokumentu.

Grafická reprezentace XQuery dokumentu je představená pomocí konstrukcí XQuery jazyka a také uzly, reprezentující XML dokumentu 4.1.

Jednotlivé uzly XQuery dokumentu jsou v grafu znázorněny takto:



Obrázek 4.1.6: sémantický graf XQuery


4.1.4 Jak začít pracovat s XQueryTool

Při prvním spuštění programu by měl uživatel nastavit pracovní adresář a umístění XML souborů. Existují dvě možnosti jak nastavit pracovní adresář:

1. Použít cestu *Menu File* ⇒ *Preferences* ⇒ *XQuery Location*.
2. Na levé stráně, kde je rozmístěn adresář souboru zmáčknout odkaz *Set folders* a vybrat příslušnou složku.

Umístění XML dokumentů lze nastavit použitím cesty jako v bodě 1. Pokud uživatel neprovede nastavení hned, může se vrátit k tomuto bodu později. Veškerá nastavení program zapamatuje a při dalším spuštění programu je zobrazí.

Pro práce s XQuery dokumenty uživatel má dvě možnosti:

Vizualizace uzlu	Popis
 <p data-bbox="327 1384 853 1444">declare function local:follows(\$a as node(), \$b as node()) as xs:boolean {...}</p> <p data-bbox="327 1460 853 1547">\$:item in doc("items.xml")//item_tuple</p> <p data-bbox="327 1570 853 1624">import schema ipo="http://www.example.com/IPO" at "ipo.xsd";</p>	<p data-bbox="906 562 1361 674">Uzel reprezentuje obecný svazek klauzulí for, let, where, order by, return.</p> <p data-bbox="906 696 1273 775">Uzel reprezentuje klauzule „FOR“.</p> <p data-bbox="906 819 1273 898">Uzel reprezentuje klauzule „LET“.</p> <p data-bbox="906 943 1273 1021">Uzel reprezentuje klauzule „WHERE“.</p> <p data-bbox="906 1055 1273 1133">Uzel reprezentuje klauzule „ORDER BY“.</p> <p data-bbox="906 1167 1273 1245">Uzel reprezentuje klauzule „RETURN“.</p> <p data-bbox="906 1267 1353 1379">Uzel reprezentuje konstrukci {}, což znamená, že potomci jsou XQuery výrazem.</p> <p data-bbox="906 1391 1289 1469">Uzel reprezentuje deklarace funkce.</p> <p data-bbox="906 1480 1345 1559">Uzel reprezentuje vnitřní část výrazu XQuery v {} závorkách.</p> <p data-bbox="906 1570 1313 1648">Samostatný uzel reprezentuje import schema.</p>

Tabulka 4.2: Vizuální uzly XQuery grafu

1. Otevřít existující XQuery dokument buď přes *Menu File* \Rightarrow *Open* (*Ctrl + O*) nebo vybrat soubor z adresáře,
2. Vyrobit nový XQuery dokument přes *Menu File* \Rightarrow *New XQuery* (*Ctrl + N*)

XQuery dokument lze editovat pomocí vestavěných funkcí v *Menu Edit* nebo použít funkce rovnou v editoru. Spouštět dotaz lze buď v *Menu XQuery* \Rightarrow *Execute* nebo zmáčknout tlačítko *Execute* v editoru nebo zmáčknout *F5*. Vyhodnocení dotazu může chvíli trvat a výsledek je zobrazen v dolní části panelu. Tento výsledek pak lze uložit nebo exportovat do různých formátů.

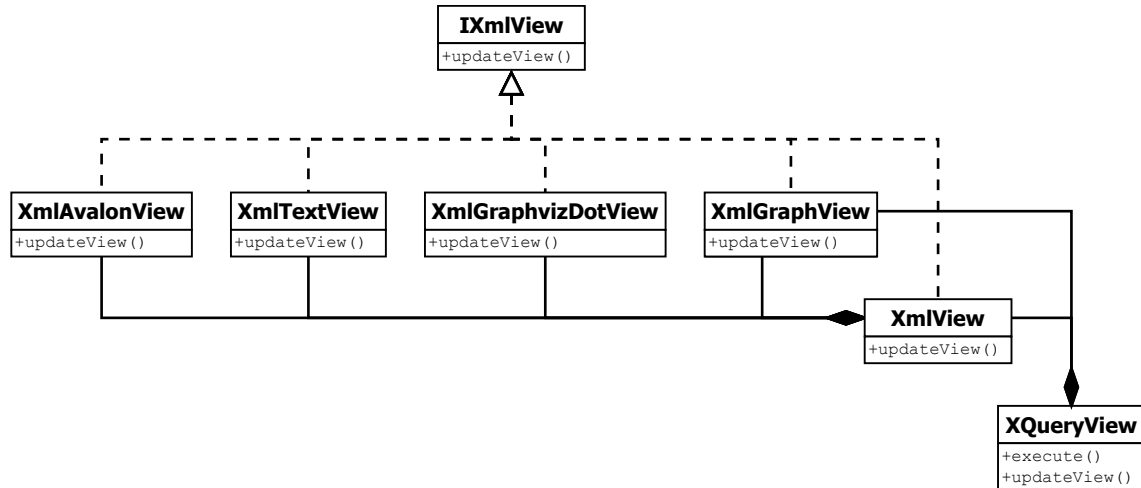
Uživatel také může pro kontrolu otevřít XML dokumenty, nad kterými dotaz probíhá.

4.2 Architektura aplikace

Klíčová část aplikace, která je zobrazená na obr.4.2.1, je kompozicí několika pohledů. Základními pohledy jsou XmlView a XQueryView.

XmlView se stará o vizualizaci XML dokumentů a zapouzdřuje několik pohledů: XmlAvalonView, XmlTextView, XmlGraphDotView, XmlGraphView.

XQueryView se stará o vizualizaci XQuery dokumentů a výsledku dotazu. Pro editaci používá Avalon editor (obdobně jako XmlAvalonView), pro vizualizaci XQuery dokumentu používá XmlGraphView z XmlView a pro zobrazení výsledku používá další XmlView.



Obrázek 4.2.1: Class diagram klíčových částí aplikace XQueryTool

4.3 XQuery v XmlGraphView

Nejzajímavější ze všech pohledů je XmlGraphView, který se stará o vizualizaci XML dokumentu ve tvaru stromu a o vizualizaci XQuery dokumentu ve tvaru sémantického grafu.

Ve jmenném prostoru `QueryMachine.XQuery` [22] jsou definovány třídy, které se zabývají analýzou a vnitřní reprezentací sémantiky XQuery dotazu.

Gramatika XQuery je popsána Backus-Naurovou normální formou (BNF), ze které je pomocí nástroje Jay generován parser `YYParse`. Tento parser je používán k převodu textové reprezentace na instanci třídy `Notation` z jmenného prostoru `QueryMachine.XQuery`.

Opačný převod umožňuje třída `XQueryWriter`, čehož využívá `XmlGraphView`. Třída `XQueryWriter` bylo doplněna o volání delegátů, kterých `XmlGraphView` využívá k sestavení grafu reprezentujícího data uložená v `Notation`. Delegáty jsou rozmístěny na různých zajímavých místech, jako je například začátek a konec elementu, začátek a konec atributu nebo i různá místa ve výrazech, což umožňuje zachytit elementy a atributy během činnosti `XQueryWriteru`.

4.4 Další vývoj aplikace

Při vývoji aplikace `XQueryTool` bylo nalezeno několik problémů, navzájem se vylučujících.

Například, není vyřešen problém vyznačení části XML dokumentů, které jsou v dotazu vybrány a jejich transformace do výstupu. Důraz byl kladen na podporu větších souborů, kvůli čemuž byla použita třída `XQueryDocument` pro reprezentaci XML dokumentu. Model `XQueryDocumentu` se skládá z objektů, které vytváří při čtení dokumentu. Každý uzel v modelu dokumentu má unikátní název a typ XSD uvnitř rodičovského uzlu. Pokus rozšířit `XQueryNavigator` a model o příznak zvýraznění se nezdařil, protože několik uzlů je reprezentováno jedním objektem a proto není možné jednoduše jim nastavovat příznak pro zvýraznění, zvýrazní to všechny uzly se stejným názvem ve stejné úrovni. Takže optimalizace použité pro minimalizaci spotřeby paměti značně komplikují rozšiřování funkčnosti.

Možné řešení spočívá v nabídnutí uživateli dvou možností: první bude pro větší soubory, ale bez vyznačení části XML dokumentu, a druhá bude omezena na velikost souboru, ale s vyznačením částí XML dokumentu s použitím objektového modelu, který načte do paměti soubor a bude s ním manipulovat.

Aplikace vizualizuje XQuery dokument jako sémantický graf, a to směrem z textu \Rightarrow graf, možné rozšíření by mohlo provázat vizualizace XQuery dotazu oběma směry, aby byla možnost vygenerovat z grafické podoby zdrojový text XQuery dokumentu.

Také by bylo možné přidat uživatelská nastavení parametrů vizualizace jednotlivých prvků.

Kapitola 5

Závěr

Cílem práce byla implementace softwaru pro podporu výuky jazyka XQuery. Aplikace XQueryTool umožňuje editovat XQuery dokumenty v textové podobě, a vizualizovat je graficky. XQuery dotazy lze vyhodnocovat, a výsledek je vizualizován textově i graficky. Výsledek lze uložit jako normální XML dokument, ale také ho lze exportovat do různých formátů, jako například PDF, SVG a další.

Pro větší přehlednost textové reprezentace XQueryTool používá obarvení syntaxe, stejné jako v grafické reprezentaci používá obarvení uzlů.

Aplikace XQueryTool byla otestovaná na příkladech k ní přiložených [14]. Tyto příklady jsou používány jako základ, uživatel může je spouštět, případně může je kombinovat a vytvářet složitější dotazy nad XML daty.

Příloha A

Obsah příloženého CD

XQueryTool.pdf – Tento text ve formátu PDF.

Thesis/ – Zdrojové soubory pro tento text ve formátu pro editor LyX.

BpVisualXQuery/ – Zdrojové kódy (úplné repositáře Gitu).

BpVisualXQuery/BpVisualXQuery/bin/Debug/BpVisualXQuery.exe – Spustitelný soubor aplikace XQueryTool.

BpVisualXQuery/BpVisualXQuery/tools/graphviz-2.28.0.msi – Instalační balík knihovny Graphviz pro kreslení grafů.

BpVisualXQuery/BpVisualXQuery/Samples – Příklady použití XQuery dotazů.

Literatura

- [1] Altova XMLSpy. <http://www.altova.com/xmlspy.html>. 2.1
- [2] AvalonEdit. <http://wiki.sharpdevelop.net/AvalonEdit.ashx>. 1.2
- [3] Document Object Model (DOM) W3C. <http://www.w3.org/DOM/>. 3.2
- [4] GraphSharp. <http://graphsharp.codeplex.com/>. 1.2
- [5] Graphviz - Graph Visualization Software. <http://www.graphviz.org/>. 1.2
- [6] <oXygen/> XML Editor version 14.0. <http://www.oxygenxml.com/>. 2.2
- [7] QuickGraph, Graph Data Structures And Algorithms for .NET.
<http://quickgraph.codeplex.com/>. 1.2
- [8] SAXON The XSLT and XQuery Processor.
<http://saxon.sourceforge.net/>. 3.3
- [9] Stylus Studio. <http://www.stylusstudio.com/>. 2.3
- [10] The World Wide Web Consortium. <http://www.w3.org/>. 1
- [11] XML Path Language (XPath) Version 1.0 W3C , November 1999.
<http://www.w3.org/TR/xpath/>. 1
- [12] XML Schema Part 0: Primer Second Edition W3C, October 2004.
<http://www.w3.org/TR/xmlschema-0/>. 1
- [13] Extensible Markup Language (XML) 1.1 (Second Edition) W3C, August 2006.
<http://www.w3.org/TR/xml11/>. 1
- [14] XML Query Use Cases W3C, March 2007.
<http://www.w3.org/TR/xquery-use-cases/>. 5
- [15] XSL Transformations (XSLT) Version 2.0 W3C, January 2007.
<http://www.w3.org/TR/xslt20/>. 3.3
- [16] Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C, November 2008.
<http://www.w3.org/TR/2008/REC-xml-20081126/>. 1

-
- [17] XML Path Language (XPath) 2.0 (Second Edition) W3C, December 2010.
<http://www.w3.org/TR/xpath20/>. 1, 3.1
- [18] XQuery 1.0: An XML Query Language (Second Edition) W3C, December 2010.
<http://www.w3.org/TR/xquery/>. 1
- [19] XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition) W3C, December 2010. <http://www.w3.org/TR/xpath-functions/>. 3.1
- [20] Standalone XQuery and SQL Implementation in .NET, November 2011.
<http://qm.codeplex.com/>. 1.2, 3.4.1
- [21] Jihad Boulos, Marcel Karam, Zeina Koteiche, and Hala Ollaic.
XQueryViz: An XQuery Visualization Tool. 2006.
<http://www.springerlink.com/content/0770186179123693/>. 2.4, 2.4
- [22] Chertkov S. A. QueryMachine.XQuery: Implementace XQuery 1.0 pro platformu .NET. June 2010.
<http://www.rsdn.ru/article/xml/QueryMachineXQuery.xml>. 1.2, 3.4, 4.3
- [23] Esposito D. : *XML - efektivní programování pro .NET*. Grada, 2004. 3.2

