Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

Бакалавърска програма
„Софтуерно инженерство”

Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2019/2020 год.

Тема №40: „Каталог на ресторантите в България“

Курсов проект

Автори:

Арина Русева, фак. номер 62207

Божидара Пачилова, фак. номер 62172

януари, 2020

София

Съдържание

1 Въведение

Този документ описва решението на задачата за съставяне на каталог на ресторантите в България по региони посредством XML технологии.

В днешно време сме свидетели на „прилив от информация“ в интернет. Когато търсим нещо, например ресторант, получаваме като резултат разнородни данни, в най-различен формат и от най-разнообразни източници. Използването на семантични технологии в интернет би допринесло за това цялата тази информация да се използва по предназначение и да бъде по-достъпна, както от хора, така и от машини. Именно тези казуси решава контекста на текущия проект, чието решение представя информация, намерена в интернет, по един семантичен начин.

За целите на проекта са приложени XML документи, валидирани чрез DTD описание, които представят основните характеристики съответно на регионите, веригите ресторанти и ресторантите, съставляващи каталога. Включено е графично съдържание, дефинирано под формата на XML Entitites. Връзките в каталога са описани чрез атрибути от тип ID/IDREF. Информацията е представена и в човеко-четим формат в PDF документ, генериран посредством XSLT (XSL-FO – Formatting Objects) транформация.

2 Анализ на решението

2.1 Работен процес

Съдържанието на каталога е представено в XML документ, като информацията за ресторантите и веригите е извлечена ръчно от източници от интернет. XML документът е валидиран чрез DTD схема. Съдържанието се представя в PDF документ чрез прилагане на XSLT трансформации и по-специално чрез технологията на Apache FOP процесор (Formatting Objects Processor), която предоставя средствата за извеждане и стилизиране на XML данни в PDF чрез съответно XSLT-FO синтаксис и CSS-like форматиране. За това генериране използвахме средата за разработка Altova XMLSpy 2020.

2.2 Структура на съдържанието

1. **restaurants-catalogue-bg** – това е кореновият елемент на схемата. Той се състои от елементите **regions, chains, price-categories** и **restaurants**, всеки от които е задължително да присъства веднъж в документа.

2. **regions** – това е пряк наследник на кореновия елемет, състоящ се от един или повече елемента **region.**

   2.1. **region** – единственото дете на елемента **regions**, който от своя страна има също единствено дете - **region-name**. Елементът има атрибута - **region_id.**

2.1.1. region-name – елемент от тип PCDATA, чието име самодокументирано описва предназначението му.

2.1.2. region_id – атрибут на елемента **region** от тип ID, отбелязан като #REQUIRED, т.е. задължителен. Чрез него еднозначно се идентифицира регион. Един регион може да бъде реферирaн от ресторантите (елементи **restaurant, т. 5.1.**) , намиращи се в него.

3. chains – пряк наследник на кореновия елемент, състоящ се от един или повече елемента **chain**.

3.1. chain – единствено дете на елемента **chains**. Състои се от елементите **chain-name, chain-description, chain-logo, chain-website,** като само последният не е задължителен, т.е. една верига може да не притежава уебсайт. Елементът **chains** има атрибут - **chain_id** от тип ID, който е задължителен и еднозначно идентифицира дадена.

3.1.1. chain-name – първото от децата на елемента **chain**, което е от тип PCDATA.

3.1.1.1. chain-description - дете на елемента chain, което е от тип PCDATA.

3.1.1.2. chain-logo – дете на елемента **chain**, което има единствено дете – **logo,** което според DTD схемата може да се сръща точно веднъж.

3.1.1.2.1. logo – това е елемент от тип EMPTY, тъй като няма поделементи и съдължание и се използва за представяне на графично изображение посредством атрибут, сочещ към източнка му. Този атрибут на елемента **logo** е **logo_src**.

3.1.1.2.1.1.logo_src - задължителен атрибут на елемента **logo** от тип ENTITY. Такъв тип атрибути сочат към ресур в различен от XML формат и в случая се използва за представяне на изображение в документа.

3.1.1.3. chain-website - дете на елемента **chain,** което е от тип PCDATA и не е задължително да присъства.

3.1.1.4. chain_id - атрибут на елемента **chain** от тип ID, отбелязан като #REQUIRED, т.е. задължителен. Чрез него еднозначно се идентифицира верига ресторанти. Една верига може да бъде реферирана от ресторантите (елементи **restaurant, т. 5.1.**), принадлежащи към нея.

4. price-categories – пряк наследник на кореновия елемент, състоящ се от един или повече елемента **category**. Предназначението на този елемент е да се обособят 4-те типични ценови категории, към които обикновено се причисляват заведенията и се визуализират под формата на символи като „$“ и други, или просто с надпис, в различните информационни източници.

4.1. category – пряк наследник на елемента **price-categories** от тип PCDATA, описващ с думи ценовата категория на ресторанта. Има два атрибут **- price_category_id**.

4.1.1. price_category_id – задължителен атрибут на елемента **category** от тип ID. Чрез него се идентифицира типът ценова категория, която ще бъде дефинирана от ресторант, посредством негов атрибут, описващ ценовата му категория.

Важно е да се отбележи също, че ценовата категория не се определя на ниво верига, а на ниво ресторант, тъй като поради различните локации на заведенията е възможно цените да варират.

5. restaurants - пряк наследник на кореновия елемент, състоящ се от поне един елемент **restaurant.**

5.1. restaurant – дете на елемента **restaurants**. Състои се от елементите **restaurant-name, main-image, address, phone-numbers, services, working-hours, cuisine, seats-capacity, website, email, menu, description** и **gallery.** От тях, елементите **services, cuisine, website, email, menu, description** не са задължителни, тъй като е възможно тази информация да липсва в източниците. Елементът **restaurant** има 4 атрибута - **restaurant_id, chain_ref, region_ref, price_category_ref,** които ще опишем в следващите под-точки.

5.1.1 restaurant_id - атрибут на елемента **restaurant** от тип ID, отбелязан като #REQUIRED, т.е. задължителен. Чрез него еднозначно се идентифицира ресторант. Един ресторант може да бъде реферирaн от региона, в който се намира или от веригата и/или ценовата категория, към които принадлежи.

5.1.2. chain_ref - атрибут на елемента **restaurant** от тип IDREFS, отбелязан като #IMPLIED, т.е. незадължителен. Чрез него се реферира веригата, от която ресторанта е част. Той е незадължителен, тъй като има ресторанти, които не са част от верига.

5.1.3. region_ref – задължителен атрибут на елемента **restaurant** от тип IDREFS. Чрез него се реферира региона, в който ресторанта се намира. Атрибута е задължителен, тъй като региона е минимална единица информация, а и според условието на задачата, ресторантите са разппределени по региони.

5.1.4. price_category_ref - незадължителен атрибут на елемента **restaurant** от тип IDREFS. Чрез него се реферира ценовата категория на ресторанта. Този атрибут не е задължителен, тъй като е възможно тази информация да липсва.

5.1.5. restaurant-name - първото от децата на елемента **restaurant**, което е от тип PCDATA.

5.1.6. main-image - елемент от тип EMPTY, тъй като се използва за представяне на графично изображение посредством атрибут, сочещ към източника му. Този атрибут **image_src**.

5.1.6.1. image-src - задължителен атрибут на елемента **main-image** от тип ENTITY. Такъв тип атрибути сочат към ресур в различен от XML формат и в случая се използва за представяне на „заглавното“ изображение на ресторант.

5.1.7. **address** – дете на елемента **restaurant,** допълнително гранулиран на елементите **city, street и building. City** е задължително да се среща поне веднъж, според DTD описанието, докато **street и building** не са.

5.1.7.1. city – елемент от тип PCDATA, дете на елемента **address**. Градът е задължително да се посочи.

5.1.7.2. street - елемент от тип PCDATA, дете на елемента **address**. Улицата не е задължително да бъде посочена.

5.1.7.3. building - елемент от тип PCDATA, дете на елемента **address.** Понякога даден ресторант се намира в някаква сграда, например мол, затова улицата сама по себе си не е достатъчна за уточняването на адреса. Този елемент може и да не присъства в описанието на даден адрес на ресторант.

5.1.8. **phone-numbers** – дете на елемента **restaurant**, състоящ се от един или повече елемента **phone-number**.

5.1.8.1. phone-number - елемент от тип PCDATA, дете на елемента **phone-numbers**.

5.1.9. **services** - пряк наследник на елемента **restaurant**, съдържащ един или повече елемента service. Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.**

5.1.9.1. service - елемент от тип PCDATA, дете на елемента **services**. Този елемент представя услуга, предоставяна от ресторант и има единствен атрибут type.

5.1.9.1.1. **type** – задължителен атрибут на елемента **service** от тип CDATA. Приемаме, че услугите могат условно да се разделят на 3 основни типа: *extra, menu и event.*

5.1.10. **working-hours** - дете на елемента **restaurant**, имащ за деца по един брой от елементите **opening-hour** и **closing-hour.**

5.1.10.1. **opening-hour** - елемент от тип PCDATA, дете на елемента **working-hours**.

5.1.10.2. **closing-hour** - елемент от тип PCDATA, дете на елемента **working-hours**.

5.1.11. **cuisine** - дете на елемента **restaurant**, имащ за дете поне един елемент **cuisine-type.** Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.**

5.1.11.1. **cuisine-type** - елемент от тип PCDATA, дете на елемента **cuisine**, описващ видовете кухня, които ресторантът предлага.

5.1.12. **seats-capacity** - елемент от тип PCDATA, дете на елемента **restaurant**, описващ броя места на ресторанта.

5.1.13. **website** - елемент от тип PCDATA, дете на елемента **restaurant.** Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.** Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.**

5.1.14. **email** - елемент от тип PCDATA, дете на елемента **restaurant.**

5.1.15. **menu** - елемент от тип PCDATA, дете на елемента **restaurant.** Предназначението на този елемент е да предоставя линк към менюто на съответния ресторант в интернет, ако такова е наличво. Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.**

5.1.16. **description** - елемент от тип PCDATA, дете на елемента **restaurant.** Според DTD документа, този елемент не е задължително да се среща в структурата на елемента **restaurant.**

5.1.17. **gallery** – пряк наследник на елемента **restaurant**, съдържащ един или повече елемента **image**.

5.1.17.1. **image -** елемент от тип EMPTY, тъй като се използва за представяне на графично изображение посредством атрибут, сочещ към източника му. Този атрибут **image_src**.

5.1.17.1.1. **image_src** - задължителен атрибут на елемента **image** от тип ENTITY. Такъв тип атрибути сочат към ресур в различен от XML формат и в случая се използва за представяне изображение от галерията на ресторант.

2.3 Тип и представяне на съдържанието

Съдържанието на каталога е представено текстово и графично в PDF формат във файла **Output.pdf**. Това представяне е осъществено посредством XSLT шаблони (templates), генериращи поредици от PDF страници (page sequences). Изплозваните мултимедийни ресурси са графични изображение в **jpg** и **png** формат. Използвани са общо **24** изображения – 11 в jpg и 13 в png формат. В приложения архив на решението те се намират в папката **images,** като са разпределени по следния начин:

- **images/logos** – съдържа логотата на веригите или на отделните ресторанти;
- **images/$restaurant-name** – съдържа снимки за отделна верига или ресторант, където **$restaurant-name** се заменя с името на веригата/ресторанта.

Изображенията и пътищата към тях са именовани така, че да се разбира какво изобразяват.

3 Дизайн

Технологиите, които използваме в нашето решение са следните:

- XML 1.0
- DTD 1.0
- XSLT 1.0
- XSL-FO
- Apache-FO Processor
- Среда Altova XML Spy 2020

Както вече стана ясно, XML документът, представящ каталога, се състои от елементи в текстов и графичен формат. Графичните елементи, както споменахме в предишната точка, са включени в папка, прилежаща към решението. В DTD схемата те са включени като ENTITY от тип NDATA с произход SYSTEM, т.е. намиращи се на локалната машина. В случая зад NDATA типа стоят нотации (NOTATION) за image (изображение) във форматите jpg и png.

В XML документа изображенията (дефинираните ENTITY-та) са включени като стойност на атрибутите ../restaurant/gallery/image**/image-src,** ../../chain/chain-logo/logo/**logo_src**, restaurant/main-image/**image-src**. За представянето им в PDF документа е използван елемента **<fo:external-graphic src="{unparsed-entity-uri(@image_src)}"/>,** който поставя така дефинираните ENTITY-та като стойност на атрибута си **src.**

Използването и описанието на връзките в каталога вече описахве в т. 2.2. Структура на съдържанието. Споменахме какво и как се реферира за всички атрибути от тип ID/IDREF. Тези атрибути са: **region/region_id, region/restaurant_ref; chain/chain_id, chain/restaurant_ref; price_category_id, price_category/restaurant_ref; restaurant_id, restaurant/chain_ref, restaurant/region_ref, restaurant/price_category_ref.**

Съдържанието на XML документа е валидирано с помощта на DTD схемата **RestaurantsCatalogue_dtd.dtd**, която е външна за **RestaurantsCatalogue.xml**, спазвайки синтаксиса и правилата на DTD, а именно:

- XML елементите се декларират чрез ELEMENT и съдържащите ги елементи както и честота им на срещане, или типа им, ако не съдържат поделементи, например:

  <!ELEMENT chain (chain-name, chain-description, chain-logo, chain-website?)>
  <!ELEMENT chain-name (#PCDATA)>

- Атрибутите на елементите се декларират чрез ATTLIST като се посочва елемента, на който принадлежат, името, типа и декларация за задължително или не включване (REQUIRED, IMPLIED):

  <!ATTLIST logo logo_src ENTITY #REQUIRED>

- Идентификаторите и референциите към тях са валидирани чрез атрибути от тип ID и IDREF:

  <!ATTLIST category price_category_id ID #REQUIRED>
  <!ATTLIST category restaurant_ref IDREFS #IMPLIED>

- Единиците (entities) са декларирани като частни външни единици:

  <!ENTITY raffy-chain-logo SYSTEM "images/logos/raffy.png" NDATA png>

Представянето на крайният вид на каталога – в PDF документ, осъществихме използвайки средата Altova XMLSpy 2020 с безплатен 30-дневен лиценз. В нея се подават XML и XSL документи и чрез XSLT и Apache FOP процесорите средата генерира PDF документ.

Структурата на XSL документа е изградена от множество шаблони, за да се разделят по-големите части от XML дървото на по-малки смислови единици, които по-лесно да могат да бъдат форматирани и поддържани.

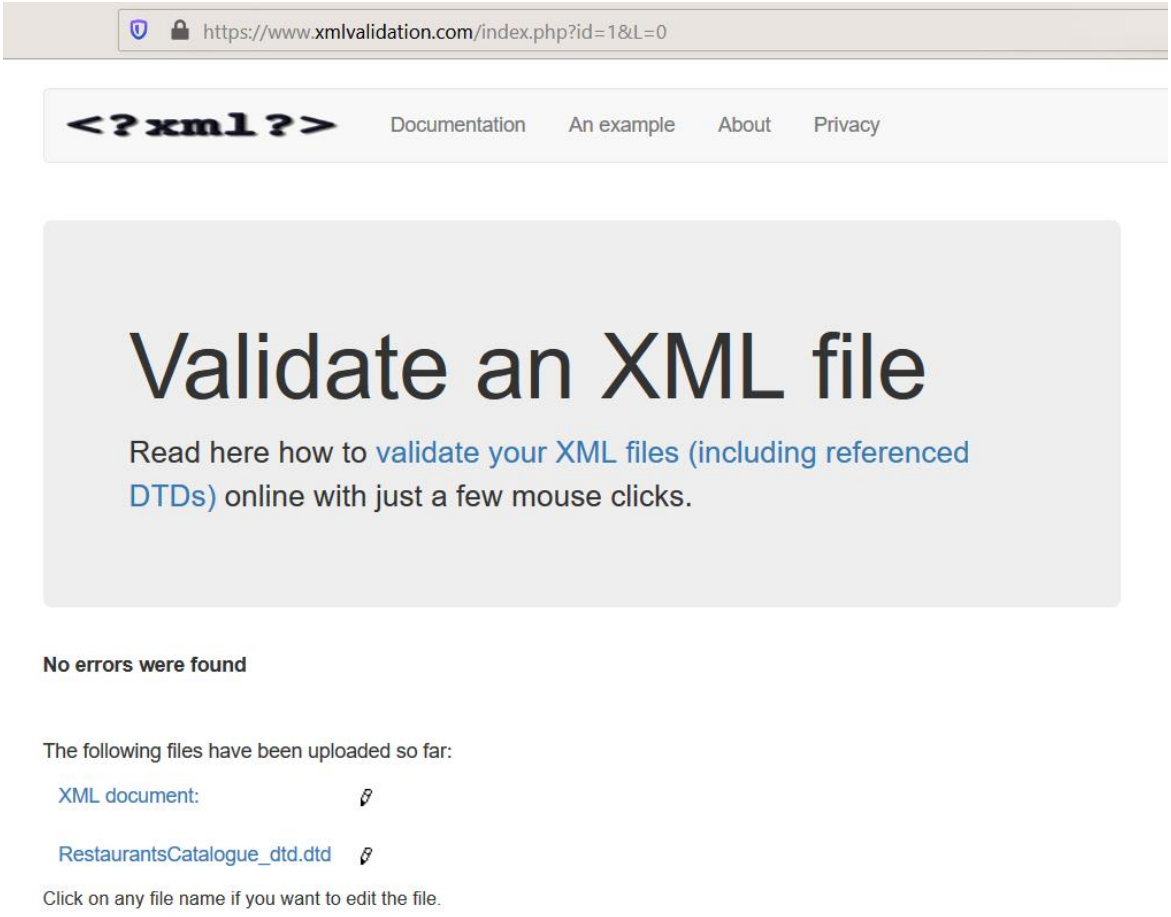
Използвайки XSL-FO синтаксисът, можем да форматираме цялостният изглед на документа чрез елементите **fo:simple-page-master, fo:region-body, fo:region-before/after** и техните атрибути **margins, page-width, page-height** и др.

Нови страници въвеждаме с елемента **fo:page-sequence.** Съдържанието в страниците се подрежда чрез множества от т.нар. **fo:block.** Експериментирали сме и с други елементи като таблица (**fo:table**, за представяне на галерията на ресторант) и списък (**fo:list-block**). Отделните самостоятелни елементи също могат да се форматират чрез CSS-подобни атрибути и стойности. За генерирането на самото съдържание от XML документа се използва стандартен XSL синтаксис.

За представяне на еднородни елементи в XSL документа използваме цикли (**xsl:for-each**) с цел намаляване на повторенията и обема, както и по-лесна възможност за промяна.

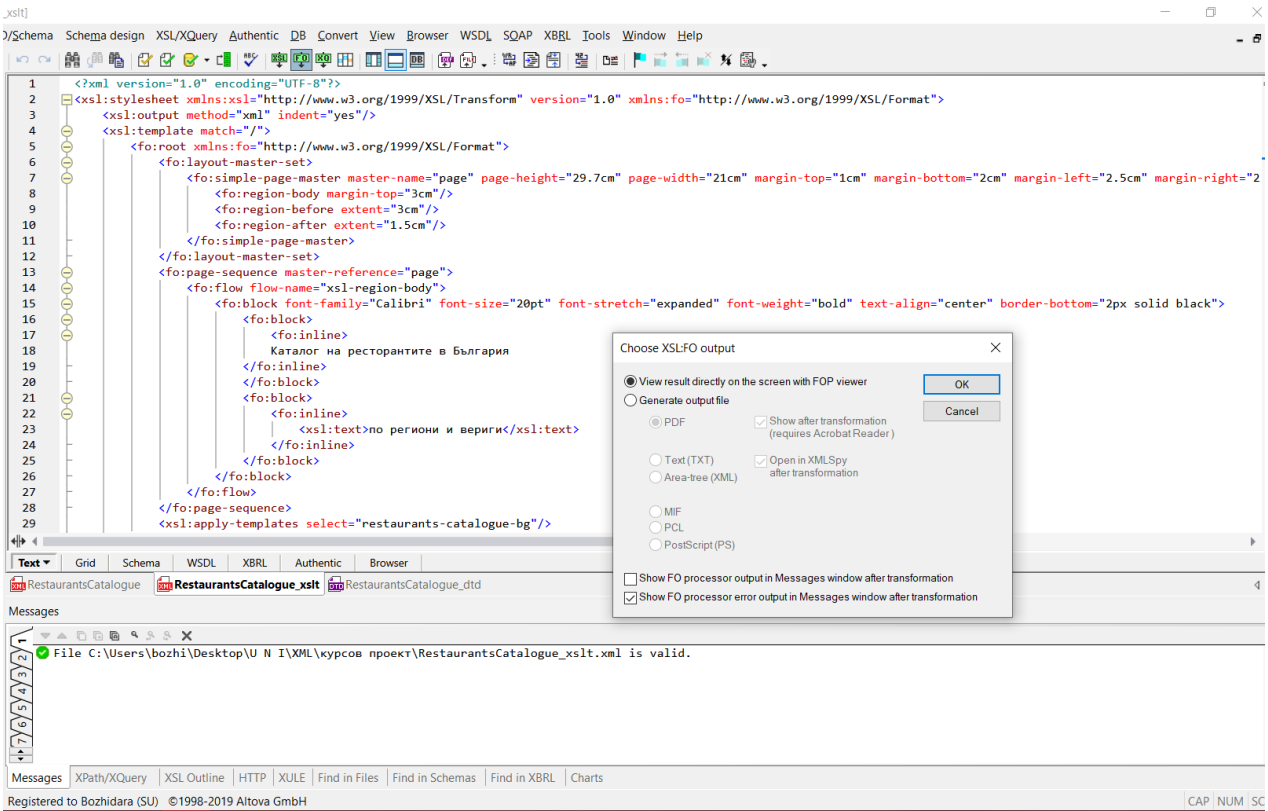
4 Тестване

XML документът е валидиран чрез DTD документ. За проверка е използван инструментът XML validator (www.xmlvalidation.com).



Фиг. 1 – Валидиране на XML документа чрез DTD схема в online среда за валидиране

За тестване на графичното представяне на XML съдържанието в PDF документ използвахме средата Altova XMLSpy 2020, която позволява лесен преглед на резултата и по време на работа



върху файла, което улеснява работния процес.

Фиг.2 – генериране на съдържанието на XML документа в PDF чрез XSLT и средата Altova XMLSpy 2020

5 Заключение и възможно бъдещо развитие

Полученият каталог представя най-важната информация, необходима за представянето на един ресторант в интернет по лесно четим както за човек (PDF), така и за машините (XML, XSLT) начин. Възможно е да се разшири с още допълнителни елементи. Хубава функционалност би била възможност за филтриране по признаци или сортиране по различни критерии (както може да се направи с ценовата категория благодарение на елемента price-category). Разбира се, това няма как да стане в рамките на PDF технологията.

Алтернатива за представянето на информацията от каталога е HTML форматът, който бихме използвали, ако искаме да представим съдържанието директно в браузър. PDF-ът, от своя страна, е подходящ за печат и по-стабилно съхранение на каталога.

6 Разпределение на работата

Арина намери информацията за ресторантите в посочените източници и дефинира DTD схемата. Божидара написа структурата на XML документа и XSLT трансформацията. Писането на документацията бе съвместно.

7 Използвани литературни източници и Уеб сайтове

Източници за ресторантите (както и за изображенията):

1. Уеб сайт на Il Siciliano - https://www.ilsiciliano.eu/restaurant
2. Уеб сайт на Raffy - http://www.raffy.bg
3. Уеб сайт на Щастливеца - http://www.shtastliveca.com/
4. Уеб сайт на Catch'a mak - https://catchamak.wixsite.com/restaurants
5. Уеб сайт на Montecito - https://www.hotelmontecito.bg/
6. Информация за различните елементи като работно време, ценова категория и т.н. взехме от страниците на съответните ресторанти в - https://www.restaurant.bg/.

Източници за технологиите:

1. Информация за синтаксиса на XSL Formatting Objects - https://www.w3.org/2002/08/XSLFOsummary.html
2. Информация за форматирането и други аспекти от XSL-FO - https://xmlgraphics.apache.org/fop/fo.html
3. Сайт на Altova XMLSpy (и източник за сваляне на програмата) - https://www.altova.com/xmlspy-xml-editor