

КОМБІНОВАНИЙ СПОСІБ МАНІПУЛЮВАННЯ ВМІСТОМ XML-ФАЙЛІВ НА ОСНОВІ XSLT ТА DTD

Вінницький національний технічний університет

Анотація

Розглянуто особливості застосування технології XSLT для перетворення структури документа та використання формату DTD для перевірки коректності документу і запропоновано спосіб їх сумісного використання для підвищення продуктивності процесу пошуку фрагменту XML-файлу.

Ключові слова: XML, XSLT-перетворення, DTD-формат, маніпулювання, DOM-дерево.

Abstract

Features of application of XSLT technology for document structure transformation and use of DTD format for checking the correctness of the document are considered and the method of their joint use is proposed for increasing the productivity of the search process for the XML file fragment.

Keywords: XML, XSLT-transform, DTD-format, manipulation, DOM-tree.

Вступ

З розвитком інформаційного суспільства все більший вплив на ефективність виробництва оказує рівень організації документообігу на підприємстві. Світовою практикою великих компаній стала організація потужного структурованого документообігу на основі кросплатформеного формату документу, який підтримується розширюваною мовою розмітки документів XML (eXtensible Markup Language). Здійснення аналізу структури і вмісту великих XML-файлів потребує спеціальних інтерактивних програмних засобів, що забезпечують пришвидшення пошуку необхідного фрагменту.

Підвищення продуктивності процесу пошуку необхідного фрагменту XML-файлу та виконання операцій із його вмістом потребує дослідження механізмів аналізу структури XML-файлів.

Модифікація застосування XSLT-перетворення

За стандартом XML кожен документ має тип, що складається з елементів. За допомогою XSLT на його основі можна створити інший документ: XML, HTML, PDF, і т. ін., що знімає необхідність двічі або тричі вносити одну і ту саму інформацію до різних документів [1].

Крім того, можна застосувати XSLT і для перетворення однієї структури на іншу, що автоматично додає до можливостей програмного засобу, заснованого на даній технології, можливість обміну даними з конфігураціями, що мають різні структури метаданих. При цьому програмні модулі завантаження та вивантаження інформації не потребують додаткових модифікацій чи закладення додаткових моделей, що є суттєвим для грамотної побудови системи документообігу, в якій усі компоненти мають бути узгодженими між собою, і містити всю необхідну інформацію, оскільки повторні введення даних підвищують можливість людських чи машинних помилок, які викликають конфлікти при міграції.

Застосуємо для створення на основі даного документа нового документа з іншою структурою. технологію XSLT (eXtensible Style Language Transformation). Для цього розіб'ємо вихідний XML-документ на окремі елементи, які формують так зване «вихідне дерево». Сценарій перетворення документу, який називається «листом стилю перетворення» (transformation style sheet), ставить у відповідність цьому дереву певне «результуюче» дерево, яке може містити всю інформацію з вихідного дерева (повне перетворення), або лише її частину (неповне перетворення). При створенні похідного документа частіше застосовується неповне перетворення, а для відображення – повне [2].

Саме це нам і потрібно, оскільки персональні дані мають відображатись у вигляді таблиці. Тепер залишилось покласти вихідний файл та сценарій перетворення на сервер, і отримати кінцевий результат (рисунки 1).



Рисунок 1 – XSLT-перетворення

Використання таких сценаріїв дозволяє позбутись надлишкових документів, що дублюють один одного, а натомість створювати нові документи "на льоту" на основі вже існуючих документів.

Формування структури елементів DOM-дерева. Ядро DOM є деревовидною структурою

XML-документу, до якої аналізатор DOM завантажує XML-документ, спрощуючи перевірки на коректність документу і роботу із ним. Загальний принцип роботи наведено на рисунку 2 [3].

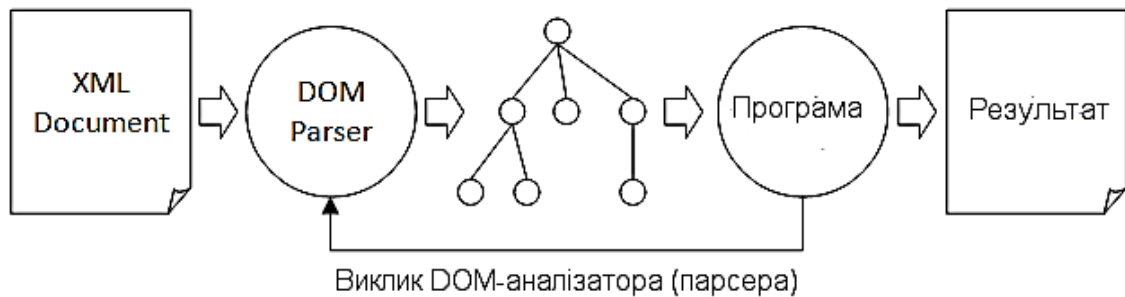


Рисунок 2 – Загальний принцип роботи DOM-аналізатора

Даний аналізатор фактично виконує три перших кроки алгоритму візуалізації – перевіряє документ на правильність, створює його деревовидну модель і завантажує до неї дані, автоматично визначаючи кореневий документ [3]. Код для застосування аналізатору є достатньо простим (рис.3).

```
FileInputStream fis = new FileInputStream(fileName);
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document document = builder.parse(fis);
```

Рисунок 3 – код для застосування DOM-аналізатору

Але візуалізації його змісту необхідним є не тільки власне подання документу, але й його деревовидне подання. Деревостворюється за допомогою рекурсивного перегляду структури і перенесення відповідних даних в елемент керування, який це дерево здатне відобразити. Розглянемо структуру типу даних Document [4].

На рисунку 4 наведено структуру типу даних Document [4], який сам по собі є деревовидною структурою, але не візуальною, а логічною. Основними вузлами структури є елементи, кожен з яких може містити атрибути або текстовий контент.

Таким чином, достатньо знайти потрібний елемент керування, який здатен відобразити подібне дерево – і задача загалом вирішена. Крім того, знадобиться табличка з двох колонок, що буде містити у собі перелік атрибутів та їх значень.

Комбінований механізм аналізу структури XML-файлів

Документом, що визначає синтаксис конкретного типу документу XML є DTD (document type definition). Документи DTD використовуються для визначення коректності документів XML, тобто їхньої відповідності структурі, визначеній документом DTD і можуть бути або вбудовані у документ XML, або знаходитися в окремому файлі. В останньому випадку документ XML містить посилання на DTD. Синтаксичні правила написання документів DTD трохи відрізняються від правил мови XML.

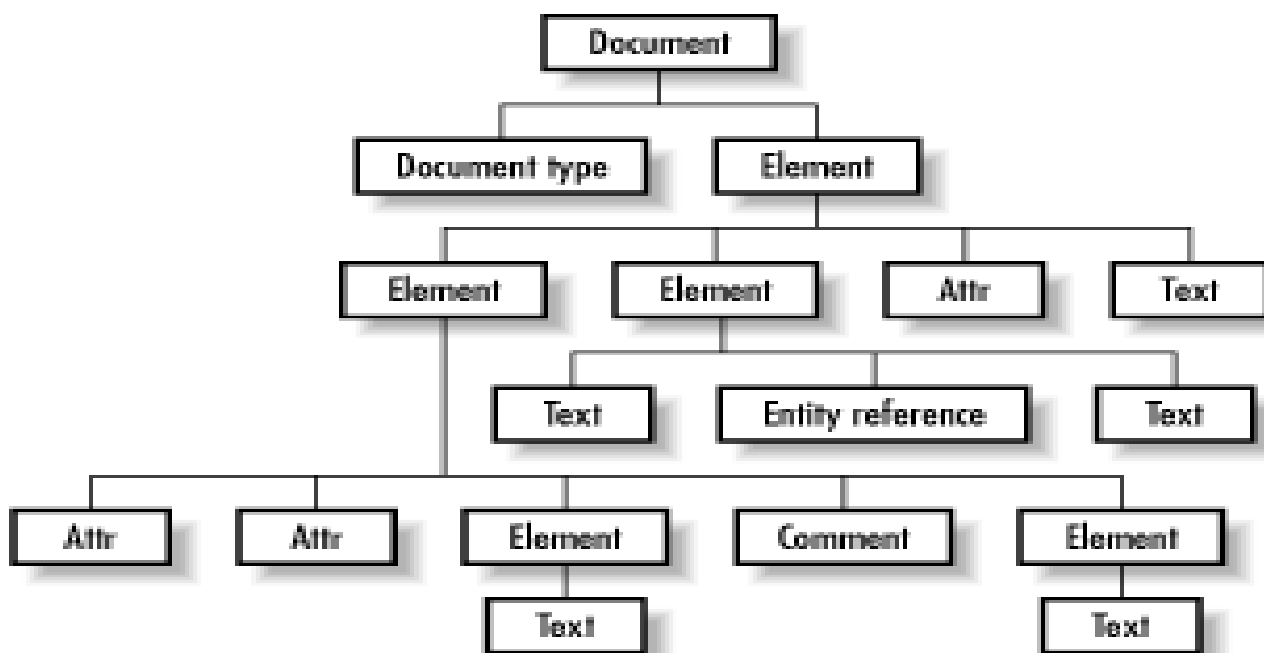


Рисунок 4 – Структура елементів типу даних DOM Document

Документ DTD не призначений для обробки аналізатором. Перший приклад адреси – це добре структурований документ, який можна прочитати і проаналізувати. Але для перевірки його коректності необхідно звірити його формат з документом DTD. Включення шаблону DTD до кожного документу XML можна замінити посиланнями на нього. Тоді аналізатор для перевірки коректності документу повинен буде одержати вихідний документ DTD або скористатися його копією з буфера, що значно підвищить продуктивність механізмів обробки даних.

Таким чином, комбінування механізмів аналізу структури XML-файлів на основі XSLT-перетворення та формату DTD дозволяє збільшити швидкість пошуку фрагменту XML-файлу.

Висновок

У роботі запропоновано комбінований спосіб розбору структури XML-файлів із можливістю маніпулювання їх вмістом, при якому застосовані механізми аналізу структури XML-файлів на основі XSLT-перетворення та формату DTD, що дозволило збільшити швидкість пошуку фрагменту XML-файлу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Язык преобразований XSL (XSLT) [Електронний ресурс] // Портал для програмистов. – Режим доступу. – <http://www.codenet.ru/webmast/xml/xslt/w3c.php>
2. XML & XSLT [Електронний ресурс] // Портал «project.net.ru». – Режим доступу: http://project.net.ru/web-master/xml_xslt/
3. Сімошенко Д. В. Інформаційна технологія маніпулювання вмістом XML-файлів / Д. В. Сімошенко // Наукові підсумки 2017 року, XV Міжнародна науково-практична Інтернет-конференція. – м. м. Вінниця, 15 грудня 2017 р. – Ч.6, С. 63-68.
4. Электронный Архив для инженеров программного обеспечения. [Електронний ресурс] // Портал «www.cs.queensu.ca». – Режим доступу: <http://www.cs.queensu.ca/Software-Engineering/>

Сімошенко Дмитро Васильович – студент групи 1 КН-16м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: simon-13@yandex.ru.

Науковий керівник: **Месюра Володимир Іванович** – к.т.н., доцент, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Dmytro V. Simonenko – Student of Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: simon-13@yandex.ru.

Supervisor: **Volodymyr I. Mesyura** – Cand. Sc. (Eng), Assistant Professor, Professor of the Computer Science Chair, Vinnytsia National Technical University, Vinnytsia.