

.NET – реализация встраиваемой XML-СУБД

77-30569/383689

04, апрель 2012

Бекасов Д. Е.

УДК 004.93'1

МГТУ им. Н.Э. Баумана

bauman@bmstu.ru

Введение

Очень часто при разработке программного обеспечения возникает задача выбора оптимальной технологии хранения данных и доступа к хранимым данным. Как правило, решением данной задачи является использование какой-либо сторонней системы управления базами данных – СУБД. На сегодняшний день такие системы реализованы под практически любые типы задач.

Распространенной является ситуация, когда использование полноценной СУБД не оправдано. А существующие встраиваемые (библиотечные) решения не подходят функционально (например, не способны в необходимой мере отобразить структуру данных) или не поддерживают необходимый язык программирования. Данная работа посвящена описанию возможного решения этой задачи: библиотеки классов под *.Net Framework*, реализующей основные функции XML-СУБД. Ниже будет введено понятие XML-СУБД и описаны основные функции разработанной библиотеки.

1 Область применения XML-СУБД

Среди большого числа современных СУБД существует один нераспространенный и специфический класс систем — XML-СУБД. Системы, относящиеся к данному классу, хранят данные в формате XML. Как правило, они предоставляют возможность хранения, обработки, создания, пересылки и упорядочивания XML-документов. XML (англ. *eXtensible Markup Language* — расширяемый язык разметки) это рекомендованный W3C (англ. *World Wide Web Consortium* – Консорциум Всемирной Паутины) язык, фактически представляющий собой свод общих синтаксических правил [1]. XML — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML). XML является упрощённым подмножеством языка SGML [2].

Все базы данных, позволяющие хранить данные в формате XML, можно разделить на две группы: XML-БД (*Native XML Database, NXD*); классические (как правило, реляционные) БД с поддержкой XML (*XML Enabled Database, XED*). Ниже перечислены основные достоинства NXD перед XED при обработке XML-данных.

- Большая простота и естественность при работе с документо-ориентированными XML, имеющими сложное или рекурсивное содержимое, сложно поддающееся реляционному отображению.
- Работа с высокоструктурированными данными (посредством использования схемы, строго задающей структуру данных).
- Возможность работы с полуструктурированными данными или данными, имеющими изменяющуюся структуру (посредством использования нескольких схем данных или не использования их вообще).
- Возможность более простого формулирования запросов к структуре XML-документа, поиска по всему документу или манипуляции задаваемой XML-документом рекурсивной моделью.
- Возможность представления XML-данных без потерь и искажений (при преобразовании в реляционную модель часть данных может исказиться, потеряться).
- Возможность использования NXD для управления потоком сообщений между частями большой распределенной информационной системы или между отдельными приложениями предприятия (интеграция между которыми часто реализуется с помощью XML-документооборота).
- XML-СУБД (в частности, встроенные) легче в создании и управлении чем серверные реляционные СУБД. Простота разработки объясняется наличием большого числа реализаций синтаксических анализаторов, языков запросов и других автоматизированных средств работы с XML на всех платформах [3].

Недостатки использования XML в качестве формата хранения и представления данных:

- хранение большого количества лишней информации;
- отсутствие типов, так как все записи в XML-файле являются строками;
- отсутствие встроенных механизмов контроля целостности;
- возможная сложность организации запросов к XML содержимому, так как это требует знания структуры каждого конкретного XML-документа;
- отсутствие индексации (поскольку XML - текстовый файл) [2].

Перечисленные проблемы, как правило, решаются при разработке XML-СУБД программно и с помощью схем XML: DTD (англ. *Document Type Definition* – определение типа документа [4]); XSD (язык описания структуры XML-документа [5]); XDR (международный стандарт передачи данных в Интернете [6]). Таким образом, сфера применения NXD ограничивается задачами хранения, обработки и представления данных сложной, иерархической и рекурсивной структуры, плохо отображающихся на реляционную модель, а также данных с изменяющейся

структурой и управления документооборотом на предприятии. Кроме этого, NXD позволяет быстро построить компактное решение для хранения и обмена данными.

2 Обзор существующих решений

NoSQL (англ. *not only SQL*, не только *SQL*) - решения, направленные на реализацию моделей баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка *SQL*. Описание схемы данных в случае использования *NoSQL*-решений может осуществляться посредством различных структур данных – хеш-таблиц, деревьев и других [7]. *NoSQL*-подход не является отрицанием языка *SQL* и реляционной модели, а скорее расширяет возможности БД и предоставляет альтернативное решение проблем там, где *SQL* недостаточно гибок. На ресурсе <http://nosql-database.org> представлено более 120 *NoSQL*-СУБД, распределенных по следующим категориям [7]:

- 1) поколоночные БД (*Wide Column Store / Column Families*);
- 2) документно-ориентированные БД (*Document Store*)
- 3) хранилища типа «ключ-значение», кортежные хранилища (*Key Value / Tuple Store*);
- 4) совместимые с хранилищами типа «ключ-значение» (*Eventually Consistent Key Value Store*);
- 5) графические БД (*Graph Databases*);
- 6) объектные БД (*Object Databases*);
- 7) сетевые и облачные БД (*Grid & Cloud Database Solutions*);
- 8) *XML*-БД (*XML Databases*);
- 9) многозначные БД (*Multivalued Databases*);
- 10) другие *NoSQL*-БД (*Other NoSQL related databases*);
- 11) вне категорий (*Unresolved and uncategorized*);

XML базы данных (категория 8) представлены решениями, указанными в таблице 1, составленной по материалам [7-14].

Таблица 1 – Обзор существующих XML-СУБД

Название	API	Протоколы	Языки запросов	Исх. код	Лицензия
<i>Mark Logic Server</i>	<i>Java, .NET</i>	<i>HTTP, REST</i>	<i>XQuery, XSLT</i>	<i>C++</i>	Коммерч.
<i>EMC Documentum xDB</i>	<i>Java, XQuery</i>	<i>WebDAV, web services</i>	<i>XQuery, XPath, XPointer</i>	<i>Java</i>	Коммерч.
<i>eXist</i>	<i>XQuery, XML:DB API, DOM, SAX</i>	<i>HTTP/REST, WebDAV, SOAP, XML-RPC, Atom</i>	<i>XQuery</i>	<i>Java</i>	Свободная
<i>Sedna</i>	<i>C, Java, Scheme, PHP, Python</i>	<i>Sedna Client-Server Protocol</i>	<i>XQuery</i>	<i>C/C++</i>	<i>Apache License</i>
<i>BaseX</i>	<i>RESTful API, WebDAV, XML:DB, XQJ, Java, C#, Perl, PHP, Python</i>	<i>BaseX Server Protocol</i>	<i>XPath XQuery</i>	<i>Java</i>	<i>BSD License</i>
<i>Qizx</i>	<i>Java</i>	<i>HTTP, REST</i>	<i>XQuery, XQuery Full-Text, XQuery Update</i>	<i>Java</i>	Коммерч. и свободная
<i>Berkeley DB</i>	<i>C/C++, Java, Perl, Python, TCL и др.</i>	<i>RPC</i>	<i>XQuery</i>	<i>C++</i>	<i>Sleepycat License</i>

Mark Logic Server. Данная СУБД предназначена для обработки больших массивов данных. Включает в свой состав масштабируемый облачный XML-сервер ёмкостью в 1 Петабайт, сервер приложений, транзакционный сервер постоянного хранения (XML, текстовые и двоичные данные) и полнотекстовую поисковую машину. Поддерживает полный набор свойств ACID (атомарность, согласованность, изолированность, долговечность [15]) и четыре вида индексации. Система может работать на кластерах с практически не ограниченным числом узлов [7, 8].

EMC Documentum xDB. СУБД поддерживает возможность одновременного чтения и записи, транзакционный контроль. Основные достоинства данной системы заключаются в поддержке полностью транзакционной модели DOM (объектной модели документа [16]), нескольких типов индексации, метаданных и не-XML данных, возможностях по управлению версиями и неограниченному горизонтальному масштабированию. В xDB предусматривается возможность

реализации полного цикла документооборота на предприятии. Валидация *XML* производится с помощью *DTD*, имеется возможность автозаполнения атрибутов [7, 9].

eXist. СУБД поддерживает одновременное чтение и запись. Клиентские *Web*-приложения для доступа к системе могут быть написаны на *XQuery* (язык запросов к данным в формате *XML* [17]) с использованием *XSLT* (язык преобразования *XML*-документов [18]), *XHTML* (семейство языков разметки веб-страниц на основе *XML* [19]), *CSS* (формальный язык описания внешнего вида документа, написанного с использованием языка разметки [20]) и *Javascript* (прототипно-ориентированный скриптовый язык программирования [21]). Поддерживает полнотекстовую индексацию, основанную на *Apache Lucene*, легкое изменение *URL* (универсальный локатор ресурса в Интернет [22]) и *XProc* (рекомендация комитета W3C по описанию *XML*-конвейеров на языке преобразования *XML* [23]). Данная СУБД хранит данные в B+-деревьях и файлах страниц, а узлы документов – в виде *DOM*-структуры, при этом хранение документов организовано иерархически. Схема индексации основана на быстрой идентификации типа структурной связи между узлами [7, 10].

Sedna. СУБД разработана и развивается отделом управления данными и информационных систем Института системного программирования РАН. *Sedna* поддерживает древовидную модель данных (храняемых в двоичном виде), которые загружаются и извлекаются в виде *XML*-документов. Данные оптимизируются и индексируются для рационального хранения и быстрого доступа. СУБД поддерживает *ACID*-транзакции, обеспечивает политику безопасности, предоставляет возможность резервирования данных, полнотекстового поиска и обновления отдельных *XML*-узлов [7, 11].

BaseX. Данная СУБД включает в себя быструю и легкую систему *XML* баз данных и *XPath* (язык запросов к элементам *XML*-документа [24])/ *XQuery* процессор с полной поддержкой последних рекомендаций W3C. Система обладает клиент-серверной архитектурой, поддерживает *ACID*, четыре типа индексации (текстовая, по атрибутам, полнотекстовая, путевая), предоставляет возможности по управлению пользователями и аутентификацией, а также включает возможность интерактивных визуализаций *XML*-документов и удобный *XQuery*-редактор реального времени [7, 12].

Qizx. СУБД предназначена для выполнения высокоскоростных запросов, поиска и обработки индексированного *XML*-содержимого. *Qizx* поддерживает параллельное чтение – запись и терабайтное масштабирование, обеспечивая при этом высокую скорость выполнения запросов. Система работает с “*well-formed*” (валидными, допустимыми) *XML* и не требует *DTD* или *XSD*. По умолчанию СУБД производит индексацию всех элементов. Библиотека *XML* реализована в виде иерархической структуры коллекций (наподобие файловой системы), содержащей сжатые данные [7, 13].

Berkeley DB XML. Система представляет из себя высокопроизводительную встраиваемую СУБД, реализованную в виде библиотеки. *Berkeley DB XML* хранит пары ключ/значение как массивы байтов и поддерживает несколько значений для

одного ключа. *BDB* может обслуживать тысячи процессов или потоков, одновременно манипулирующих базами данных размером в 256 терабайт на разнообразном оборудовании под различными операционными системами. СУБД обеспечивает гибкое хранение *XML* (документ или узел), группировку содержимого по контейнерам, валидацию по *XSD*, поддержку метаданных в формате ключ-значение, поддержку пространств имен, возможность отладки *XQuery*-запросов, гибкую индексацию *XML* [7, 14].

Из данного короткого обзора самых популярных *XML*-СУБД можно выделить основные тенденции на рынке рассматриваемых систем:

- использование *XQuery* и его модификаций как основного языка запросов;
- использование *Java* и *C++* как языков разработки СУБД;
- наличие хорошо проработанного интерфейса под популярные языки программирования;
- наличие транзакционного контроля, реализация свойств *ACID*;
- реализация нескольких видов индексации (полнотекстовая, по атрибутам, по пути);
- обеспечение иерархического хранения документов;
- представление данных с помощью *DOM*, использование *DTD* и *XSD* для валидации;
- использование клиент-серверной архитектуры СУБД.
-

3 Предлагаемое программное решение

Целью данной работы не было создание полноценной клиент-серверной СУБД для обработки данных. Основной задачей являлась разработка библиотеки классов под *.Net Framework*, реализующей основные функции встраиваемой *XML*-СУБД:

- управление базами данных;
- добавление новых документов в базу;
- основные операции с данными;
- обеспечение целостности;
- транзакционный контроль;
- индексация данных.

В качестве технологии обработки и доступа к *XML*-данным была выбрана *LINQ to XML* в силу следующих причин:

- структура *XDocument* (объектная модель документа от компании Microsoft), удобнее и нагляднее в использовании, чем *DOM*;
- использование проверяемых на этапе компиляции *LINQ*-выражений (в отличие от непроверяемых строк *XPath*);
- *SQL*-подобный синтаксис *LINQ*;
- *LINQ* – современная, развивающаяся и перспективная технология доступа к данным *.Net Framework* [25].

Библиотека классов состоит из трех основных частей.

Управление базами данных. Эта часть реализована классом *DBMS*. Данный класс при создании загружает конфигурационную информацию из *XML*-файла «*config.xml*», в которой указывается имя файла, содержащего список баз данных. Система загружает этот файл и таким образом получает список баз данных, в котором указываются файлы описания каждой базы данных. Кроме того, данный класс предоставляет методы загрузки баз данных из списка и добавления новой базы данных, а так же содержит ссылку на загруженную в данный момент базу данных.

Управление текущей базой данных. Эта часть реализована классом *Storage*. В её функции входит добавление файлов в базу данных, удаление файлов из базы данных, обеспечение целостности данных при запросах на изменение, удаление и добавление, непосредственное направление запросов пользователя к конкретным файлам базы данных, загрузка всех файлов базы данных в память.

Управление XML-документами. Эта часть реализована классом *XMLFile*. Данный класс является надстройкой над *LINQ*-классом *XDocument*, включая в себя помимо экземпляра *XDocument*, экземпляр *XMLSchemaSet* (набор схем файла), экземпляр *XDocument* (шаблон документа, выстроенный по загруженной схеме – для автоматизации процесса добавления данных), таблицу ссылок (на какие файлы данный файл ссылается) и индексную хеш-таблицу. Данный класс реализует методы загрузки, сохранения *XML*, проверки его допустимости по схеме документа, методы обработки и выборки.

Ниже рассмотрены реализации основных функций перечисленных частей.

В силу неопределенности структуры *XML* и существования разных типов запросов, реализовано несколько перегрузок метода *select*, которые в зависимости от входных данных реализуют разную функциональность. Основные перегрузки представлены в таблице 2.

Таблица 2 – Перегрузки метода выборки данных

Аргументы запроса	Возвращаемое значение
ID и список полей	Выборка указанных полей из элемента, выбранного по ID из индексной таблицы
ID	Выборка по ID из индексной таблицы
Список полей	Соединяются запросы по выборке элементов и атрибутов из списка по их первичному ключу, а результат запроса помещается в перечисление экземпляров анонимного типа
Список ID	Создается перечисление результатов запроса по каждому ID
Пусто	Возвращается документ целиком. Пользоваться этим запросом не рекомендуется, так как в этом случае теряется контроль над целостностью данных. Позволяет пользователю составлять собственные <i>LINQ</i> -запросы к документу

В запросе может быть указан конкретный узел объектной модели документа, по которому надо выполнять выборку. В этом случае метод применяется не к целому документу, а к его части.

При вызове метода обновления указывается список подзапросов обновления, каждый из которых содержит имя обновляемого элемента, первичный ключ, данные для обновления, путь к элементу (с помощью перечисления идентификаторов подэлементов, если первичный ключ не указывает путь к элементу однозначно), хеш-таблицу атрибутов для обновления, функцию обновления, определяемую тип запроса. Предусмотрены следующие типы запроса обновления: изменение данных, вставка новых подэлементов, удаление данных, добавление данных и тип, определяемый пользователем. В зависимости от указанного типа, делегату функции обновления (структура данных, указывающая на методы экземпляра класса в .NET Framework) присваивается та или иная лямбда-функция (анонимная функция, которая содержит выражения и операторы и может использоваться для создания делегатов).

Удаление элемента происходит по индексу и реализовано средствами класса *XDocument*. При вызове метода добавления к корню *XML*-документа добавляется новый дочерний элемент, созданный по шаблону. После этого во вновь созданном элементе устанавливаются все значения, указанные в запросе (при этом, аналогично ситуации с обновлением, может быть указан путь).

Для контроля целостности при загрузке файлов базы данных, система создает хеш-таблицы списков ссылок. Для каждого файла базы данных в этой таблице указывается список файлов, которые на него ссылаются. Контроль целостности осуществляется при возникновении одного из двух событий, описанных ниже.

- При удалении элемента. При этом проверяется, не ссылаются ли на него из других файлов. Для этого по имени файла из таблицы ссылок извлекается список файлов, ссылающихся на данный и каждый файл из списка спрашивается на предмет наличия ссылок на конкретный элемент (т.е. содержит ли он заданный внешний ключ). Если такие файлы есть, то генерируется соответствующее исключение.
- При обновлении и добавлении элемента. В этих случаях проверяется, не указывается ли в запросе внешний ключ, ссылающийся на несуществующий элемент. В этом случае так же генерируется соответствующее исключение.

Для повышения производительности системы при любых операциях выборки данных реализована возможность индексирования по первичному ключу. Индексирование осуществляется путем составления хеш-таблицы по первичному ключу. Значениями хеш-таблицы являются узлы загруженного в память *xml*-документа.

В данной системе за одну транзакцию принимается одна операция. Она либо применяется целиком, либо отменяется. Это достигается тем, что все манипуляции производятся с заранее созданной копией исходного документа, которой исходный документ замещается только в случае успешности всех действий. Под операцией понимается последовательность подзапросов обновления, соединенных в один пакет или запрос вставки элемента.

Заключение

В статье определена область применения XML баз данных, рассмотрены существующие No-SQL решения, описана архитектура разработанной встраиваемой XML-СУБД. Предлагаемое решение представляет собой библиотеку классов на языке C# и поддерживает необходимый набор функций – загрузку и сохранение XML-БД, выборку, добавление, изменение и удаление данных, обеспечение целостности, реализацию индексации данных по первичному ключу, транзакционный контроль.

Список использованных источников

- 1 XML Technology // W3C. URL. <http://www.w3.org/standards/xml/> (дата обращения 05.03.12).
- 2 XML // Wikipedia.URL.<http://ru.wikipedia.org/wiki/XML> (дата обращения 05.03.12).
- 3 Leigh Dodds. XML и базы данных// XMLhack.ru: Новости XML-технологий <http://www.xmlhack.ru/texts/02/followNose/followNose.html> (дата обращения 05.03.12).
- 4 XML Specification DTD//W3C.URL.<http://www.w3.org/XML/1998/06/xmlspec-19980910.dtd> (дата обращения 05.03.2012).
- 5 XML Schema Part 0: Primer Second Edition//W3C.URL.<http://www.w3.org/TR/xmlschema-0/> (дата обращения 05.03.2012).
- 6 XDR: External Data Representation Standard//IETF.URL.<http://www.ietf.org/rfc/rfc4506.txt> (дата обращения 05.03.2012).
- 7 LIST OF NOSQL DATABASES//NOSQL.URL.<http://NoSQL-database.org> (дата обращения 05.03.12).
- 8 MarkLogic 5//MarkLogic. URL. <http://www.marklogic.com/products-and-services/marklogic-5/> (дата обращения 05.03.12).
- 9 Documentum xDB//EMC2.URL.<http://www.emc.com/products/detail/software2/documentum-xdb.htm> (дата обращения 05.03.12)
- 10 Overview//eXist-db Open Source Native XML Database.URL. <http://exist-db.org/> (дата обращения 05.03.12).
- 11 About Sedna//Sedna:native xml database system. URL.<http://www.sedna.org/> (дата обращения 05.03.12).
- 12 BaseX. The XML Database.//BaseX.URL. <http://basex.org/>(дата обращения 05.03.12).
- 13 Qizx, a native XML database engine fully supporting XQuery//XMLmind: the XML Products Division of Pixware.URL.<http://www.xmlmind.com/qizx/>(дата обращения 05.03.12).
- 14 Oracle Berkeley DB XML//Oracle.URL.<http://www.oracle.com/us/products/database/berkeley-db/xml/overview/index.html/>(дата обращения 05.03.12).
- 15 Gray J. The Transaction Concept: Virtues and Limitations // Proceedings of the 7th International Conference on Very Large Databases – 1981. – Вып. – 144. – URL. <http://research.microsoft.com/en-us/um/people/gray/papers/theTransactionConcept.pdf> (дата обращения 21.02.2012).

- 16 Document Object Model (DOM)//W3C.URL.<http://www.w3.org/DOM/> (дата обращения 05.03.2012).
- 17 XQuery 1.0: An XML Query Language (Second Edition)//W3C.URL.<http://www.w3.org/TR/xquery/> (дата обращения 05.03.2012).
- 18 XSL Transformations (XSLT) Version 1.0 //W3C.URL. <http://www.w3.org/TR/xslt> (дата обращения 05.03.2012).
- 19 XHTML 1.0 The Extensible HyperText Markup Language (Second Edition) //W3C.URL. <http://www.w3.org/TR/xhtml1/> (дата обращения 05.03.2012).
- 20 Cascading Style Sheets, level 1//W3C.URL. <http://www.w3.org/TR/CSS1/> (дата обращения 05.03.2012).
- 21 ECMAScript Language Specification //Ecma International.URL. <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf> (дата обращения 05.03.2012).
- 22 Uniform Resource Locators (URL)//IETF.URL.<http://tools.ietf.org/html/rfc1738> (дата обращения 05.03.2012).
- 23 XML Processing Model Working Group//W3C.URL.<http://www.w3.org/XML/Processing/> (дата обращения 05.03.2012).
- 24 XML Path Language (XPath) Version 1.0//W3C.URL.<http://www.w3.org/TR/xpath/> (дата обращения 05.03.2012).
- 25 LINQ//Библиотека msdn. URL. <http://msdn.microsoft.com/en-us/library/bb397926.aspx> (дата обращения 05.03.12).

A .NET implementation of embedded XML-DBMS

77-30569/383689

04, April 2012

Bekasov D.E.

Bauman Moscow State Technical University

bauman@bmstu.ru

The place of the XML-DB was determined in this article, existent No-SQL solutions were considered, the architecture of the developed system was also described. The proposed solution is a class library in C# and supports a set of functions, such as loading and saving XML-database, selecting, inserting, updating and deleting the data, data integrity, indexing on the primary key, transaction control. The application domain of this system is .Net-applications working with small amounts of data with a complex structure. Further development of this work is an addition of full-text indexing and an implementation of security policies.

Publications with keywords: [XML](#), [.Net](#), [C#](#), [NoSQL](#), [XML-DBMS](#), [XSD](#), [LINQ](#), [embedded database](#)

Publications with words: [XML](#), [.Net](#), [C#](#), [NoSQL](#), [XML-DBMS](#), [XSD](#), [LINQ](#), [embedded database](#)

References

1. *XMLTechnology.W3C*. Available at: <http://www.w3.org/standards/xml/>, accessed 05.03.12.
2. *XML. Wikipedia*. Available at: <http://ru.wikipedia.org/wiki/XML>, accessed 05.03.12.
3. *LeighDodds. XML i bazy dannykh. XMLhack.ru: Novosti XML-tekhnologii* [LeighDodds. XML and databases. News of XML technology]. Available at: <http://www.xmlhack.ru/texts/02/followNose/followNose.html>, accessed 05.03.12.
4. *XML Specification DTD. W3C*. Available at: <http://www.w3.org/XML/1998/06/xmlspec-19980910.dtd>, accessed 05.03.2012.
5. *XML Schema Part 0: Primer Second Edition. W3C*. Available at: <http://www.w3.org/TR/xmlschema-0/>, accessed 05.03.2012.

6. *XDR: External Data Representation Standard*. IETF. Available at: <http://www.ietf.org/rfc/rfc4506.txt>, accessed 05.03.2012.
7. *LIST OF NOSQL DATABASES*. NOSQL. Available at: <http://NoSQL-database.org>, accessed 05.03.12.
8. *MarkLogic 5*. MarkLogic. Available at: <http://www.marklogic.com/products-and-services/marklogic-5/>, accessed 05.03.12.
9. *Documentum xDB*. EMC2. Available at: <http://www.emc.com/products/detail/software2/documentum-xdb.htm>, accessed 05.03.12.
10. *Overview. eXist-db Open Source Native XML Database*. Available at: <http://exist-db.org/>, accessed 05.03.12.
11. *About Sedna. Sedna: native xml database system*. Available at: <http://www.sedna.org/>, accessed 05.03.12.
12. *BaseX. The XML Database*. BaseX. Available at: <http://basex.org/>, accessed 05.03.12.
13. *Qizx, a native XML database engine fully supporting XQuery. XMLmind: the XML Products Division of Pixware*. Available at: <http://www.xmlmind.com/qizx/>, accessed 05.03.12.
14. *Oracle Berkeley DB XML*. Oracle. Available at: <http://www.oracle.com/us/products/database/berkeley-db/xml/overview/index.html/>, accessed 05.03.12.
15. Gray J. The Transaction Concept: Virtues and Limitations. *Proceedings of the 7th International Conference on Very Large Databases*. 1981, Iss. 144. Available at: <http://research.microsoft.com/en-us/um/people/gray/papers/theTransactionConcept.pdf>, accessed 21.05.2012.
16. *Document Object Model (DOM)*. W3C. Available at: <http://www.w3.org/DOM/>, accessed 05.03.2012.
17. *XQuery 1.0: An XML Query Language (Second Edition)*. W3C. Available at: <http://www.w3.org/TR/xquery/>, accessed 05.03.2012.
18. *XSL Transformations (XSLT) Version 1.0*. W3C. Available at: <http://www.w3.org/TR/xslt>, accessed 05.03.2012.
19. *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*. W3C. Available at: <http://www.w3.org/TR/xhtml1/>, accessed 05.03.2012.
20. *Cascading Style Sheets, level 1*. W3C. Available at: <http://www.w3.org/TR/CSS1/>, accessed 05.03.2012.
21. *ECMAScript Language Specification*. Ecma International. Available at: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>, accessed 05.03.2012.

22. *Uniform Resource Locators (URL)*. IETF. Available at:
<http://tools.ietf.org/html/rfc1738>, accessed 05.03.2012.
23. *XML Processing Model Working Group*. W3C. Available at:
<http://www.w3.org/XML/Processing/>, accessed 05.03.2012.
24. *XML Path Language (XPath) Version 1.0*. W3C. Available at:
<http://www.w3.org/TR/xpath/>, accessed 05.03.2012.
25. *LINQ. Biblioteka msdn* [LINQ. Msdn library]. Available at:
<http://msdn.microsoft.com/en-us/library/bb397926.aspx>, accessed 05.03.12.