

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ XML В DELPHI

В.А. Парасич

APPLICATION OF XML TECHNOLOGIES IN DELPHI

V.A. Parasich

Рассмотрена XML-технология для описания смыслового содержания документов, указаны ее основные преимущества и особенности. Описаны несколько технологий обработки документов XML в среде разработки Delphi.

Ключевые слова: XML-технология, обработка документов, среда Delphi.

XML-technology describing document storage content is considered; its basic advantages and features are emphasized. Several methods of XML documents processing in programming environment DELPHI are described.

Keywords: XML technology, documents processing, programming environment Delphi.

В настоящее время Интернет рассматривается как средство обмена бизнес-документами в электронном формате. Для этой цели был сформирован набор стандартов, основанный на применении формата XML. В этот набор вошли такие стандарты, как транспортный протокол SOAP, схемы XML, предназначенные для проверки корректности документов, а также технология XSL, позволяющая преобразовать документы XML в формат HTML [1].

В последнее время также много внимания уделяется построению систем электронного бизнеса, или, как их еще называют, B2B (business to business). Учитывая рекомендации по построению обменных потоковых систем координирующего интернет-технологии органа – WWW Consortium, акцент сделан в сторону XML-технологий и построения систем обмена XML-документами [2].

Особенностью XML является то, что с помощью тегов (определенной системы разметки) он позволяет описывать смысловое содержание документа. При этом язык XML не имеет фиксированного набора тегов. Их можно свободно придумывать для собственных нужд, расширяя возможности системы. Хотя имеются зарезервированные наборы тегов для конкретных областей применения, ничто не мешает использовать те же названия в собственных приложениях и для других целей.

Сегодня язык XML поддерживают практически все ведущие компании, производящие программное обеспечение, поэтому реализация этого языка имеется и в системе Delphi.

Преимущество использования XML в электронном бизнесе – высокая эффективность B2B систем при низких затратах на ее создание за счет четкого и наглядного представления структурированной информации, возможность использования современных сетевых протоколов и создания бизнес-систем реального времени.

Независимость представления информации в виде XML-документов позволяет разным, участвующим в электронном бизнесе, фирмам производить независимое друг от друга программное обеспечение.

Во всех системах обмен, как правило, строится по одинаковой схеме, с использованием HTTP-запросов. В качестве протокола защиты информации применяется, как правило, протокол SSL.

Одним из возможных вариантов обработки XML-сообщения является построение BIN/CGI-приложений (ISAPI) или COM-компонент (серверных), формирующих или обрабатывающих XML-документы.

Приложение, с одной стороны, выступает в качестве клиента, которое в режиме POST выдает HTTP-запрос, с другой – находится на WEB-сервере, на стороне которого осуществляется обработка запроса и выдача ответа. В информационном обмене используются XML-документы.

Один из наиболее эффективных вариантов реализации – использование существующего XML-парсера, поддерживающего DOM-модель. Такой парсер является дистрибутивной поставкой ОС Windows98 и более поздних версий или составной частью InternetExplorer практически всех

Парасич Виктор Александрович – канд. техн. наук, доцент кафедры электронно-вычислительных машин, Южно-Уральский государственный университет; pva16@yandex.ru

Parasich Viktor Alexandrovich – Candidate of Science (Engineering), Associate Professor of the Department of Electronic Computers, South Ural State University; pva16@yandex.ru

применяемых версий и представляет COM-сервер, находящийся в библиотеке MSXML.DLL.

Модель компонентных объектов (COM) объединяет инкапсулированные данные и методы в единую сущность и осуществляет способ доступа к ним через систему интерфейсов. Средствами Delphi достаточно просто осуществить доступ к классам COM-объекта (в одном COM-сервере может быть включено несколько классов). Доступ к объектам осуществляется путем инициализации экземпляра класса через систему интерфейсов. Описание интерфейсов осуществляется языком определения интерфейсов (IDL), которое возможно осуществить средствами среды автоматически.

Существует несколько технологий обработки документов XML, но основными являются применение SAX (Simple API for XML) и использование DOM (Document Object Model). Рассмотрим отличия этих подходов [1].

SAX (Simple API for XML), не загружая документ в оперативную память, самостоятельно осуществляет грамматический разбор XML-кода и генерирует события для каждого его логического элемента. Когда грамматический разбор документа посредством SAX завершается, документ становится в системе недоступным и дальнейший доступ к нему без повторной загрузки невозможен. Однако операция грамматического разбора с использованием SAX выполняется значительно быстрее, чем построение в памяти древовидной структуры DOM. Поэтому применять SAX удобнее в случае, когда требуется обработать XML-документ однократно, например, если в нем выполняется поиск некоторого элемента данных.

DOM (Document Object Model) загружает весь документ в память в виде иерархического дерева узлов. После этого есть возможность получить доступ к любому из них и выполнить в отношении него некоторые процедуры, включая операции редактирования. Поэтому DOM удобнее применять в том случае, если необходимо неоднократно перемещаться по структуре XML-документа и редактировать ее. Дополнительно посредством DOM можно с нуля создать новый XML-документ. В этой технологии каждый XML-документ представляется в виде набора множества объектов (классов), с помощью которых возможен доступ к отдельным элементам (полям объекта). DOM-интерфейс описывает доступ как к простым объектам типа DOMString или CharacterData, так и к частям или отдельным элементам XML-документа: DOMFragmentElement, DOMNode, DOMElement.

Достаточно часто применяется третий способ создания и обработки XML-документов, основанный на применении стандартных механизмов работы со строками. Самый быстрый способ генерации XML-кода – это формирование документа при помощи добавления к нему новых строк, особенно в случае, если требуется выполнить только один

проход по документу и нет надобности в дальнейшей модификации уже сгенерированных узлов. Следует также учесть, что чтение XML-документа с использованием строковых функций осуществляется очень быстро.

Помимо упомянутых стандартных механизмов, доступных и в других языках программирования, среда разработки Delphi поддерживает еще две дополнительные технологии работы с документами XML. Обе эти технологии доступны только в Delphi.

Первая дополнительная технология предусматривает формирование специальных интерфейсов в соответствии с внутренней структурой XML-документа. Эти интерфейсы формируются исходя из структуры конкретного документа и предназначаются для работы с данным конкретным документом, на основе которого они сформированы. То есть в отличие от универсального интерфейса DOM, который может использоваться для работы с любыми документами XML, интерфейсы, специально сформированные средой Delphi, предназначены для обработки одного конкретно взятого документа и не могут использоваться совместно с любыми документами XML. Такой подход позволяет за короткое время разработать более понятный исходный код, снижается вероятность ошибок программирования, а полученное в результате приложение обладает большей надежностью.

Второй дополнительный подход, предлагаемый в рамках Delphi, предусматривает использование трансформаций, при помощи которых содержимое документа XML можно разместить в компоненте ClientDataSet или, наоборот, сохранить содержимое компонента ClientDataSet в файле XML с заданной структурой.

Возможность преобразования наборов данных из различных внутренних форматов (например, таблиц баз данных) в стандартизованный формат XML (и наоборот) очень полезна. Такие средства позволяют стыковать работу приложений, созданных разными производителями, и расширять функциональные возможности унаследованных (морально устаревших, но активно эксплуатируемых) систем, не меняя их исходных текстов, а просто добавляя утилиты и программы конвертации данных и делая эти данные общедоступными с помощью стандарта XML. Для визуального преобразования наборов данных из внутренних форматов в XML и обратно предназначена утилита XmlMapper.

Еще одна полезная возможность набора компонентов Delphi – их способность работать с XML-файлами как с источниками данных. Такие файлы представляют собой обычный текст с тегами разметки, структурирующими содержимое не только в реляционном, но и, например, в иерархическом виде. При этом надо учитывать, что полная открытость XML-файлов делает их беззащитными от внешнего просмотра, поэтому вряд ли разумно

хранить в них какую-либо конфиденциальную информацию. И, конечно, если сервер СУБД выполняет множество дополнительных функций по ведению базы данных (поддерживает транзакции, блокирует при необходимости таблицу или запись от конфликтных изменений, сохраняет ее целостность, выполняет различные действия по оптимизации запросов), то работа с XML-файлами таких возможностей не даст. Лучше всего задействовать XML-файлы в качестве своеобразной небольшой локальной базы данных, сведения в которой хранятся не реляционно, а в виде иерархических структур. Чаще всего потребность в обработке XML-файлов востребована в интеграционных

приложениях, когда данные из одних баз и систем передаются в другие и нуждаются во временном хранилище, в идеале имеющем иерархическую структуру [3].

Литература

1. Кэнту, М. *Delphi 2005. Для профессионалов* / М. Кэнту. – СПб.: Питер, 2006. – 912 с.
2. *Использование XML в среде Delphi*. – <http://www.interface.ru/home.asp?artId=4694>
3. Бобровский, С.И. *Технологии Delphi 2006. Новые возможности* / С.И. Бобровский. – СПб.: Питер, 2006. – 288 с.

Поступила в редакцию 25 мая 2012 г.