

МЕТОДЫ ИНТЕГРАЦИИ СЕТЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ СОЧЕТАНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЙ И ЯЗЫКОВ РАЗМЕТКИ ТЕКСТА

ЕВСЮКОВ А.Ю.

(Компьютерные науки)

Интеграция средств и методов объектно-ориентированного программирования с последними достижениями в области разметки текстов позволяет внести усовершенствования в архитектуру сетевых информационных систем. Предложен один из способов подобной интеграции, основанный на использовании информации о структуре объектов.

На сегодняшний день World Wide Web (WWW) представляет собой колоссальное хранилище информации, которое основано на множестве синтаксически структурированных HTML страниц, ссылающихся друг на друга и на файлы данных самых разнообразных форматов. Быстрое развитие спецификации HTML (и WWW в целом) обусловлено правилами обработки разметки текста, позволяющими игнорировать неизвестные элементы. Целью введения такого способа работы с неизвестной разметкой было предоставить группам разработчиков и производителям программного обеспечения возможность экспериментировать с новыми языковыми элементами, обеспечивая частичную совместимость с предыдущими версиями Internet браузеров. Однако каждое новое расширение, предложенное одним из производителей, в большинстве случаев не могло быть корректно обработано альтернативными браузерами, потому что разметка текста в HTML оторвана от его семантики и, в силу этого, из-за нечеткости в определении правил игнорирования для каждого нового элемента разметки. Таким образом, каждое новое свойство HTML может широко применяться только после утверждения его в очередной версии языкового стандарта.

Отсутствие достаточных средств для структурной организации, расширяемости, проверки правильности электронных документов приводит к тому, что инфраструктура Web всё чаще не может удовлетворить современные нужды электронного издательства, корпоративных информационных систем и систем электронной коммерции. Для удовлетворения растущих потребностей коммерческого использования Web и возможности дальнейшего расширения Web-технологий в новых областях распределенной обработки документов World Wide Web Консорциум разработал Extensible Markup Language (XML) для приложений, которые требуют функциональных возможностей, выходящих за рамки Hypertext Markup Language (HTML) - формата, доминирующего в настоящее время в Internet. По-другому складывается

ситуация в сфере распределенных объектных систем.

Технология объектно-ориентированного программирования (ООП) явилась первым шагом на пути разрешения проблем, связанных с созданием сложных программных систем. Использование объектно-ориентированной технологии было вызвано необходимостью повысить производительность труда разработчиков путем повторного использования кода, стремлением строить системы из уже отлаженных блоков, потребностью более точного отражения предметной области в терминах, применяющихся в деятельности предприятия. Однако отсутствие единой системы, позволяющей объектам, разработанным различными производителями, взаимодействовать в составе программного комплекса (работающего не только в пределах адресного пространства, но и в масштабах сети), ограничивает преимущества применения ООП. Решение этой проблемы современная программная индустрия ищет в построении систем компонентного программного обеспечения, которые призваны перевести объектную технологию на более высокий уровень. Компонентное программирование стало важным достижением объектно-ориентированной технологии. Оно позволило повторно использовать объектно-ориентированный код более эффективно, так как основой для этого стал выполняемый код, а не исходные тексты или библиотеки. Компонентные модели, такие как CORBA, COM, JavaBeans, особенно бурно развиваются в сфере корпоративных распределенных вычислений. Такого рода системы имеют строго определенные интерфейсы, что, устраняя проблемы взаимодействия, приводит к тому, что любое расширение интерфейса требует достижения соглашения между всеми сторонами информационного обмена. В таких системах обе стороны, обменивающиеся информацией, обязаны «знать» структуру классов и объектов друг друга. И, соответственно, даже малые изменения на одной из сторон влекут изменения, направленные на поддержание совместимости во всей системе. Следует отметить, что существующие стандартные методы сериализации компонентов, т.е. сохранения структурно-эквивалентного представления данных приложения, обладают следующими недостатками:

- они не устойчивы к изменениям в структуре классов, породивших данный поток сериализации;
- в формате хранения отсутствует семантическая информация о структуре объектов или она жестко связана с конкретной средой разработки или языком программирования;
- организация ссылок между объектами приводит к сохранению всего, на что ссылается

данный объект, или она связана с дополнительными усилиями;

- отсутствуют стандартные механизмы для фрагментации формата представления или поддержка подобной фрагментации требует дополнительного программирования.

Кроме того, интеграция разнородных объектных систем всегда связана с созданием промежуточного слоя, предназначенного для обеспечения совместимости и адаптирующего различные спецификации на уровне форматов данных, соглашений о вызовах и платформенных различий, что влечет за собой потери эффективности.

Из сказанного выше можно сделать вывод, что в настоящее время необходима более богатая базовая модель данных, которая позволяла бы совмещать развивающиеся средства структуризации информации, представленной в Internet, с последними достижениями объектно-ориентированных технологий. Но даже сейчас эти две технологии идеологически близки. Web, например, может рассматриваться как распределенная система объектов, в которой их состояния сохраняются в виде HTML страниц, идентификация основана на URL, а поведение определяется выполнением сервером HTTP запросов. Объектная технология может послужить естественным средством добавления функциональности к статическому представлению в Web. В настоящее время Web страницы уже используются для доставки исполняемого кода в виде Java апплетов или объектов ActiveX на клиентские машины. В некоторых приложениях объекты, доставленные таким образом и выполняемые на клиенте, связываются с объектами на сервере, используя один из распределенных объектных протоколов обмена (например, OMG IIOP или Java RMI). Аспекты динамического изменения (т.е. поведения) Web страниц являются основой разработки Dynamic HTML (динамического HTML), проводимой Microsoft and Netscape. Разработка Document Object Model (объектной модели документов), проводимая W3C, направлена на развитие этой идеи в целях создания объектной модели, которая бы позволяла представлять HTML и XML документы в виде коллекции программируемых объектов. При этом код, выполняемый на клиенте, манипулирует состоянием этих объектов, изменяя состояние отображаемого документа.

Сочетание достоинств современной объектно-ориентированной технологии в виде наличия четко определенных интерфейсов с возможностью представления в данных семантически значимого, расширяемого и кросс-платформенного формата на основе XML документов, по мнению автора, позволит организовать взаимодействие разнородных систем, имеющих только общий словарь предметной области.

Такое сочетание видится в организации сериализации компонентов в виде XML документов. Предлагаемый подход обладает следующими достоинствами:

- XML допускает иерархические структуры любой вложенности и хорошо подходит для хранения объектно-ориентированных структур;

- документы этого формата несут в себе семантическую информацию о структуре данных, что позволяет создавать документы, ориентированные на использование терминов предметной области и не связанные с какой-либо системой (языком) программирования;

- XML представляет собой текстовый файл, что существенно облегчит отладку и стыковку систем, а также их сопровождение;

- XML является стандартом W3C и используется многими ведущими производителями, что обеспечивает поддержку как на уровне развития стандарта, так и на уровне обеспечения инструментами разработки.

Модель сохранения компонентов на основе разметки текста должна основываться на следующих принципах:

- механизм сохранения должен быть устойчив к изменению структуры классов;

- приложение должно обладать возможностью пропускать неизвестные элементы разметки или обрабатывать их специальным образом.

В настоящее время существует несколько разработок в области сериализации объектно-ориентированных данных в XML. Большинство из них в той или иной мере удовлетворяют перечисленным выше принципам. Проблема устойчивости к изменению набора свойств классов легко решается благодаря самой структуре формата, которая позволяет коду, взаимодействующему с кодом разбора XML, пропускать неизвестные элементы разметки, найденные во время прохода по файлу.

Однако явным недостатком существующих разработок является принцип построения кода формирования вывода в формате XML. Как правило, каждый сохраняемый класс имеет метод или группу методов, отвечающих за формирование представления свойств объекта в терминах разметки текста. В то же время код десериализации построен на использовании анализатора (parser), т.е. класса или группы взаимодействующих классов, ответственных за разбор XML. Подобная схема организации программы показана на рис. 1. Десериализация происходит следующим образом. Анализатор (XML Parser), разбирая документ, взаимодействует с кодом приложения (application code), который, в свою очередь, отвечает за создание

(блок Factory) и конфигурацию объектов классов данных (data classes) на основе данных, полученных от анализатора. Сериализация же производится самими классами данных, объекты которых записывают своё внутреннее представление непосредственно в формат XML.

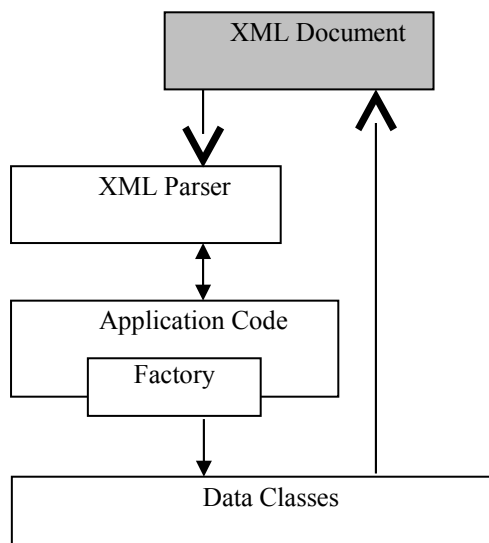


Рис. 1. Традиционная схема сериализации

Недостатками подобной схемы являются:

- необходимость внесения модификаций во многих местах программы при изменении в структуре классов предметной области;
- связь с форматом представления организована во многих местах программы, что делает ее весьма чувствительной к его изменениям, так как модификация формата влечет за собой обширное изменение кода на всех уровнях.

Кроме того, классы данных помимо своих основных функций, т.е. моделирования предметной области, несут в себе функции вывода. Такая «перегруженность» функциональности представляется не концептуальной.

Таким образом, отмечая перечисленные выше недостатки, можно сформулировать дополнительные требования к интеграции ООП и языков разметки текста:

- код сохранения компонентов в XML должен быть локализован. В идеале не нужно требовать написания специального кода сохранения или он должен быть минимален и предназначен для обработки специфических ситуаций;
- для обеспечения более гибкой связи между элементами разметки текста и классами ассоциации между ними должны, по возможности, устанавливаться вне программного кода, т.е. во внешних документах.

Эта задача может быть решена путем использования Run-Time Type Information или информации периода выполнения.

Современные объектно-ориентированные языки программирования предоставляют возможность получения информации о структуре объектов в период выполнения программы. Run-Time Type Information (RTTI), или информация о типах в период выполнения, позволяет определять класс объекта, перечислять его свойства, устанавливать их тип и менять их значение. С помощью средств RTTI отображения объектов на разметку в формате XML можно построить обмен объектно-ориентированными данными, более устойчивый к изменениям в представлении проблемной области. Использование механизма RTTI позволит также избавиться от написания кода сохранения. Предлагаемое решение для сериализации в XML показано на рис. 2.

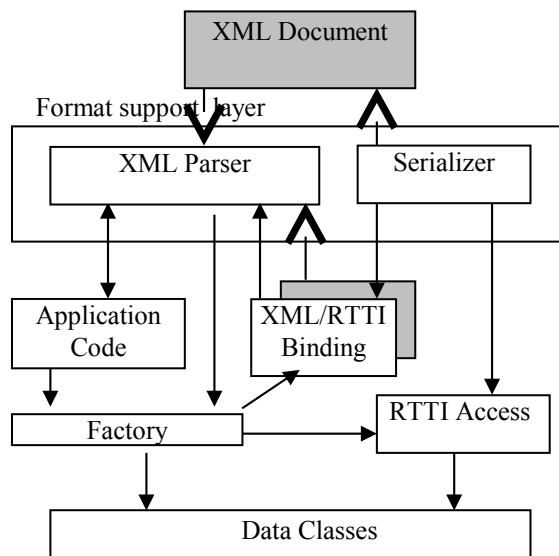


Рис. 2. Схема сериализации с использованием RTTI

Центральными элементами данной схемы являются следующие модули.

Модуль сериализации (Serializer) представляет собой сеть классов, реализующих интерфейсы одной из стандартных спецификаций для работы с XML: Document Object Model или Simple API for XML (SAX). Этот модуль взаимодействует с двумя следующими.

Модуль доступа к RTTI представляет собой сеть классов, которая инкапсулирует стандартные средства к информации времени выполнения и предоставляет дополнительные сервисные функции.

Модуль связывания RTTI информации и разметки текста (XML/RTTI Binding) включает в себя как файлы привязки (binding files), так и классы для работы с ними. Привязка осуществляется на основе внешних XML файлов, которые отображают элементы разметки на классы и их свойства. Этот

модуль взаимодействует с анализатором XML для получения информации о разметке.

Полная схема взаимодействия при десериализации выглядит следующим образом. Код приложения устанавливает связь между XML Parser и блоком создания и конфигурации объектов (Factory). После этого XML Parser и Factory взаимодействуют напрямую. Анализируя заданный документ, XML Parser посылает сообщения Factory. Factory, принимая обработанные элементы разметки текста, получает информацию о необходимых действиях из XML/RTTI Binding, который хранит отображение во внутренних структурах данных, полученных при обработке Binding файла на начальном этапе инициализации системы. Получив необходимые данные для данного элемента разметки, Factory обращается к модулю RTTI Access для создания или конфигурации объекта.

При сериализации модуль Serializer получает информацию о сохраняемом объекте через модуль RTTI Access. Элементы разметки текста, соответствующие классу данного объекта, Serializer получает из блока связывания и формирует вывод в формате XML.

Как показано на рис. 2, XML Parser и Serializer формируют уровень поддержки формата (Format support layer) и только эти два блока обрабатывают все операции, связанные с внутренним представлением потока сериализации. Такая структура позволяет локализовать изменения, связанные с форматом, на уровне одной подсистемы, не затрагивая остальной код программы. Кроме того, изменения в представлении предметной области вызывают только модификацию классов данных и файлов привязки. Следовательно, изменения формата или классов данных локализованы на уровне подсистем, т.е. не влекут за собой изменения в других подсистемах. Следует отметить, что предложенная архитектура позволяет легко менять формат сериализации. Такой переход осуществляется заменой классов в блоках XML Parser и Serializer на альтернативные, реализующие те же интерфейсы, а также заменой внешних файлов связывания на альтернативные файлы в новом формате. При наличии нескольких реализаций таких унифицированных интерфейсов и их динамическом связывании на этапе выполнения смена формата может осуществляться без перекомпиляции системы и даже во время ее выполнения. Таким образом, предложенная модель может служить основой для построения систем с гетерогенным информационным обменом, при котором общение с различными источниками информации осуществляется в разных форматах, а ядро системы позволяет подключать обработчики новых форматов и не зависит от формата внешнего представления данных.

Литература: 1. *Bosak J.* XML, Java, and the Future of the Web // XML.com. XML: Principles, Tools and Techniques. Edited by. Dan Connolly and Rohit Khare, – 1997. Oct 02. 2. *F.Manola.* Towards a Web Object Model, Object Services and Consulting, Inc. (OJS). 1998. 3. Capturing the State of Distributed Systems with XML by Rohit Khare and Adam Rifkin. XML special issue of the World Wide Web Journal, Autumn 1997, Vol. 2, Number 4, P. 207-218.

Сведения об авторе: **Евсюков Александр Юрьевич**, аспирант филиала кафедры ИИИС ХТУРЭ. Научные интересы: объектно-ориентированные технологии, компонентные модели, Internet технологии. Адрес: 310145, Харьков, ул. Новгородская 20, кв. 54, тел. 45-51-32, e-mail: aye@bigfoot.com

УДК 007.52;519.7

МЕТОДЫ ИНТЕГРАЦИИ СЕТЕВЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ СОЧЕТАНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЙ И ЯЗЫКОВ РАЗМЕТКИ ТЕКСТА / А.Ю.Евсюков // Радиоэлектроника и информатика. – 1998. – №. – с. – .

В работе описывается один из способов интеграции преимуществ объектно-ориентированного программирования и достижений Internet технологий в области разметки текстов для построения сетевых информационных систем.

Библиогр.: 3 назв.

UDC 007.52;519.7

Methods for Integration of Information Systems Based on Object-oriented Technologies and Text Markup Languages / A.Evsukov // Radioelektronika i informatika – 1998. – N . – P. – .

The paper discusses one of the methods for integration of advantages of object-oriented programming achievements of Internet technologies in the field of text markup languages.

Ref.: 3 items.