

РАЗРАБОТКА ПОЛНОФУНКЦИОНАЛЬНЫХ ВЕБ-ПРИЛОЖЕНИЙ НА ОСНОВЕ БИБЛИОТЕКИ TAUCODE

А.С. Коваленко, К.С. Коваленко

Институт кибернетики имени В.М. Глушкова НАН Украины,
Украина, 03680, МСП, Киев-187, проспект Академика Глушкова, 40

Описанная библиотека TauCode предназначена для разработки современных «богатых» веб-приложений, включая поддержку бизнес-логики на стороне сервера и клиента, конфигурирование используемых ресурсов с помощью XML, структурированную обработку исключений, компоненты графического интерфейса, поддержку асинхронного взаимодействия с сервером и технологию AJAX в целом.

The TauCode library is meant for development of up-to-date rich web applications, including server-side and client-side business-logic support, XML-based configuring of used resources, structured exceptions handling, graphics user interface components, support of asynchronous interaction with server and support of the AJAX technology in general

Введение

Библиотека TauCode разработана для обеспечения быстрого проектирования и разработки веб-приложений на всех этапах их жизненного цикла. Как правило, современное веб-приложение включает три основных составляющие: сервер приложений (Application Server), осуществляющий бизнес-логику, хранение и обработку данных; веб-сервер, служащий для взаимодействия клиента с сервером, и собственно клиент, традиционно являющийся обыкновенным веб-браузером на клиентском компьютере. Библиотека TauCode предоставляет компоненты, позволяющие быстро и качественно создавать все эти составляющие. Серверная часть библиотеки реализована на языке C#.NET, клиентская – на языке сценариев JavaScript, поддерживаемом всеми современными веб-браузерами.

СУБД

В качестве БД можно использовать любую реляционную базу данных, поддерживающую .NET Framework, в том числе такие распространенные, как Microsoft SQL Server, Oracle, MySQL, Postgres и другие. TauCode подразумевает реализацию бизнес-логики посредством написания C#-кода, а не, например, посредством кодирования ее триггерами и хранимыми процедурами СУБД, хотя позволяет использование и такой технологии.

Сервер приложений

Основой для сервера приложений является абстрактный класс `TauCode.Middleware.Server.Engine`, сборки `TauCode.Middleware.Server`, который содержит базовые методы управления сервером: пуск/остановка сервера, авторизация пользователей и администраторов сервера приложений, доступ к базе данных. TauCode рассчитана главным образом на СУБД MS SQL Server, но может работать и с любым другим сервером баз данных, для которого существует провайдер ADO.NET. При проектировании сервера приложений для своей задачи, разработчик создает класс-наследник `TauCode.Middleware.Server.Engine`, в котором переопределяет виртуальные методы в соответствии со своей предметной областью.

Предметная область

Для описания предметной области служит сборка `TauCode.Middleware.Model`, основными классами которой являются: `TauCode.Middleware.Model.BizEntity`, описывающий бизнес-сущность, `TauCode.Middleware.Model.ModelLogic`, описывающий компоненты модели, универсальные для данной предметной области, и `TauCode.Middleware.Model.EntityManager`, служащий для сериализации / десериализации экземпляров сущностей предметной области.

Методы работы с бизнес-сущностями (перечисление, создание, изменение, удаление и т. д.) реализуются посредством наследования интерфейсов `TauCode.Middleware.ICommonHandle` (операции, не требующие авторизации), и `TauCode.Middleware.IUserHandle` (операции пользователя). Интерфейс `TauCode.Middleware.IAdminHandle` используется для администрирования сервера приложений и, при необходимости, для доступа к низкоуровневым ресурсам сервера (базы данных, файлы и т. д.).

Передача данных

Для передачи данных между сервером приложений и веб-сервером может использоваться один из трех типов транспорта, поддерживаемых .NET Remoting: IPC (межпроцессное взаимодействие), TCP (передача по локальной сети), HTTP (передача посредством протокола HTTP). Также, существует возможность создавать сервер приложений в контексте веб-сервера. В таком случае, сервер приложений называется DLL-сервером, по аналогии с терминологией COM. Для организации передачи данных используется класс `TauCode.Middleware.Connectivity.RemoteConnector` (для независимых, или EXE-серверов приложений) и `TauCode.Middleware.Connectivity.InProcConnector` (для создания сервера приложений в контексте веб-сервера).

© А.С. Коваленко, К.С. Коваленко, 2010

Веб-сервер

Библиотека TauCode рассчитана на применение веб-сервера Microsoft Information Server (IIS) версии 6.0 или выше под управлением операционной системы Windows Server 2003 или выше. Для функционирования библиотеки, необходимо наличие установленного компонента ASP.NET.

В данной работе рассматриваются вопросы создания богатых приложений на основе платформы .NET Framework, веб-сервера Microsoft Information Services (IIS) и библиотеки компонентов TauCode.

Поддомены

Библиотека TauCode предполагает использование поддоменов как средство разделения различных компонентов приложения. Например, пусть проект назван foo.com. Тогда он может иметь такие компоненты:

- Система расчета: <http://accounting.foo.com>
- Блог: <http://blog.foo.com>
- Форум: <http://forum.foo.com>
- Справочная система: <http://help.foo.com>

и так далее.

Для каждого из этих компонентов выводится класс-наследник `TauCode.Web.App`, в котором переопределяются абстрактные (виртуальные) методы, отвечающие за различные аспекты функционирования компонента: генерация страниц приложения, конфигурация приложения (включая подключения к требуемым ресурсам: базы данных, файлы и т.д.), доступ к веб-ресурсам и т.д.

Множественные конфигурации

В библиотеке TauCode разработаны средства, позволяющие создавать множественные конфигурации для веб-приложений, предназначенные для различных стадий разработки. Например, набор конфигураций может содержать конфигурацию для разработки, конфигурацию для тестирования, конфигурацию для работы (production configuration). Выбор необходимой конфигурации осуществляется приложением, исходя из места расположения. Так, например, для конфигурации разработки задается доменное имя `foo.dev`, для конфигурации тестирования – `foo-test.com`, для production-конфигурации – `foo.com`. Приложение при запуске (фактически, при первом обращении клиента-браузера) выбирает нужную конфигурацию, анализируя параметр `HOST` HTTP-запроса.

Разумеется, для поддержки конфигураций разработки с доменными именами типа `foo.dev`, необходимо соответствующим образом сконфигурировать веб-сервер IIS и изменить файл `%WINDIR%\System32\drivers\etc\hosts`.

Преимуществом такого подхода является отсутствие необходимости иметь разные файлы конфигураций для разработки, тестирования и работы – все конфигурации находятся в одном файле, что уменьшает ошибки при установке (deployment) приложения на реальный (рабочий) сервер.

Конфигурация приложения описывается с помощью XML и может содержать сложные вычисляемые параметры, позволяющие производить «тонкую» настройку программным способом. Доступ к конфигурации осуществляется классом `TauCode.Tools.XmlTuning`.

Семейства страниц (Family Pages)

Для поддержки страниц, объединенных общей функциональностью (например, фронтальные веб-формы, пользовательские веб-формы, администраторские веб-формы), разработано понятие семейства страниц. Их объединяет использование совместных ресурсов (JavaScript и CSS), а также генератор страниц (`PageRenderer`).

Семейства страниц описываются при помощи XML-файлов, в которых указываются совместно используемые ресурсы, а также ресурсы, свойственные каждой отдельной странице (веб-форме). При этом, семейства страниц могут использовать ресурсы сторонних сборок, например, в случае, если многие компоненты приложения используют одни и те же ресурсы.

Семейства страниц в TauCode представлены классом `TauCode.Web.AppPageFamily`.

Асинхронное взаимодействие (AJAX)

Для обеспечения работы приложения без необходимости перезагружать страницу (веб-форму) используется технология AJAX (Асинхронный JavaScript и XML).

AJAX-запросы обслуживаются обработчиком (Generic Handler) `DoAjaxAction.ashx`, который, в свою очередь, создает экземпляр класса-наследника `TauCode.Web.AjaxAction`; обработка запроса осуществляется переопределенным методом `RunInternal()`, который осуществляет необходимые действия (десериализация запроса, обращение к серверу приложений, сериализация результата и отправка его серверу), а также, при возникновении исключения (Exception), сериализует информацию об исключении и доставляет его клиенту (рис. 1).

Веб-ресурсы

Под веб-ресурсами подразумевается JavaScript-сценарии, каскадные таблицы стилей (CSS), или фрагменты HTML-кода, используемые в том числе для темплейтов диалогов. Таким образом, веб-ресурсы – это данные, которые доставляются клиенту в добавок к основной веб-странице (веб-форме в терминах ASP.NET).



Рис. 1

Сборки ресурсов (Resource Assemblies)

Сборка ресурсов – это набор ресурсов, объединенных неким общим признаком, определяемым разработчиком. Например, имеет смысл включать в одну сборку ресурсы, используемые данным семейством страниц, или ресурсы, общие для всех компонентов приложения.

Пучки (Bundles)

В целях уменьшения сетевого трафика, в TauCode предусмотрена возможность объединения произвольного набора веб-ресурсов в так называемый пучок, генерируемый на стороне сервера и доставляемый клиенту в виде одного ресурса.

Отладка веб-сервера

Приложения на основе TauCode предназначены для разработки и отладки в среде Visual Studio версии 2005 или 2008. Встроенный в Visual Studio веб-сервер для разработки (Development Web Server) обладает ограниченной функциональностью для работы с много-доменными приложениями. В частности, в свойстве HttpRequest.Url.Host он всегда возвращает 127.0.0.1, что исключает корректную обработку подобных запросов средствами TauCode.

Поэтому для разработки приложений TauCode используется полноценный сервер IIS, который входит в поставку (в качестве устанавливаемого компонента) в операционные системы Windows XP, Windows Vista, Windows 7, и всех серверных версий Microsoft Windows.

При этом используется метод присоединения к процессу (Attach to a Process), реализованный в Visual Studio. Перед этим необходимо прописать хосты в файле %WINDOWS%\System32\drivers\etc\hosts, и добавить соответствующие хосты в настройках IIS.

Клиент

На стороне клиента, приложение представляет собой набор взаимосвязанных компонентов, написанных на языке сценариев JavaScript (рис. 2), и состоящий из:

- управляющий код – набор функций, обеспечивающий взаимодействие всех компонентов приложения;
- ядро AJAX – подсистема обработки асинхронных запросов;
- бизнес-логика – описание сущностей и деталей бизнес-модели;
- графический интерфейс – компоненты взаимодействия пользователя с приложением;
- обработчик исключений.

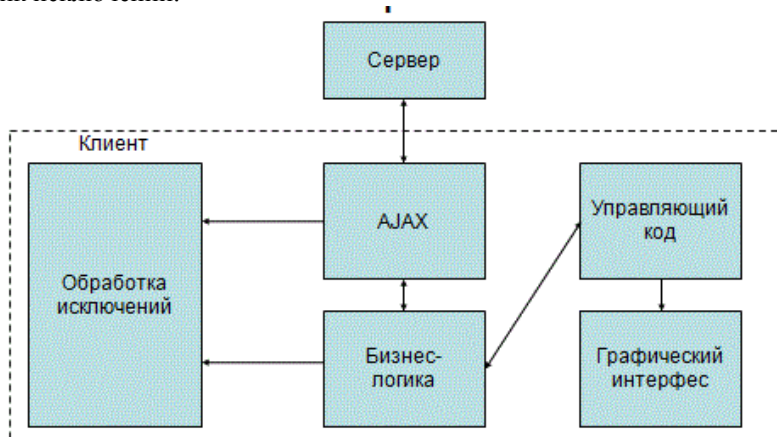


Рис. 2

Клиентская часть библиотеки TauCode поддерживает следующий функционал:

- средства взаимодействия с веб-сервером посредством асинхронных запросов;
- компоненты графического интерфейса;
- средства генерации и разбора XML-документов;
- средства поддержки арифметики дат;
- расширенная обработка строк;
- средства динамической генерации клиентского (JavaScript) кода;
- средства валидации введенных данных;
- обработку исключений;
- поддержку бизнес-логики: сущности, коллекции и взаимодействие между ними;
- средства оптимизации сетевого трафика путем объединения отдельных скриптов библиотеки в единый пакет;
- средства отладки и трассирования;
- кросс-браузерность: библиотека корректно работает на всех современных веб-браузерах.

Кроме того, библиотека обеспечивает бесконфликтную работу клиентского приложения при использовании библиотек сторонних производителей, например таких как Prototype, MooTools, jQuery и других.

Поддержка асинхронных запросов

Практически каждое современное веб-приложение поддерживает асинхронные (без обновления загруженной веб-страницы) вызовы между клиентом (браузером) и веб-сервером. Цикл взаимодействия включает в себя:

- формирование данных, которые надо отправить на сервер, и упаковка их в XML- или текстовый документ;
- отправка данных на сервер посредством встроенного объекта XMLHttpRequest;
- асинхронное получение отклика от сервера, путем использования функции обратного вызова (механизм callback);
- анализ полученного ответа и отображение их в браузере клиента.

Данный функционал реализован в библиотеке TauCode в виде (статического) класса AjaxAction, методы которого выполняют все перечисленные задачи.

Компоненты графического интерфейса

Библиотека TauCode включает все распространенные компоненты пользовательского графического интерфейса, в том числе: таблицы (grids), деревья, модальные и встроенные диалоги, календари, контекстные меню, страницы со вкладками (tab views). Все эти компоненты реализованы в виде отдельных классов, каждый из которых реализован в отдельном скрипте (JavaScript-файле), что обеспечивает высокую модульность (логическое разделение). Такое разделение позволяет избежать ошибок на этапе разработки и упрощает кастомизацию компонентов (возможность реализовывать дополнительный функционал в классах-наследниках компонентов путем использования псевдо-виртуальных методов).

Кроме того, в библиотеке реализована поддержка взаимодействия с объектами Adobe Flash, что позволяет интегрировать компоненты графической визуализации, основанные на Flash ActionScript, включая компоненты библиотек Adobe Flex, AmCharts и других (см. рис. 3–5).

Дата ▼	Сумма	Валюта	Счет	
02.12.2009	60.09	UAH	Наличные (UAH)	Удалить
02.12.2009	12.47	UAH	Наличные (UAH)	Удалить
02.12.2009	17.03	UAH	Наличные (UAH)	Удалить
02.12.2009	35.50	UAH	Наличные (UAH)	Удалить
01.12.2009	56.29	UAH	Наличные (UAH)	Удалить
01.12.2009	35.00	UAH	Наличные (UAH)	Удалить
01.12.2009	10.50	UAH	Наличные (UAH)	Удалить
01.12.2009	200.00	UAH	Наличные (UAH)	Удалить
01.12.2009	21.99	UAH	Наличные (UAH)	Удалить
Всего:	448.87	UAH ▼		

Рис. 3

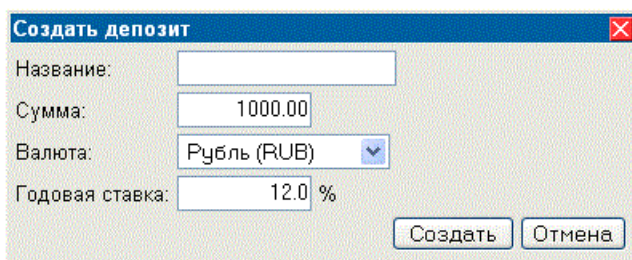


Рис. 4

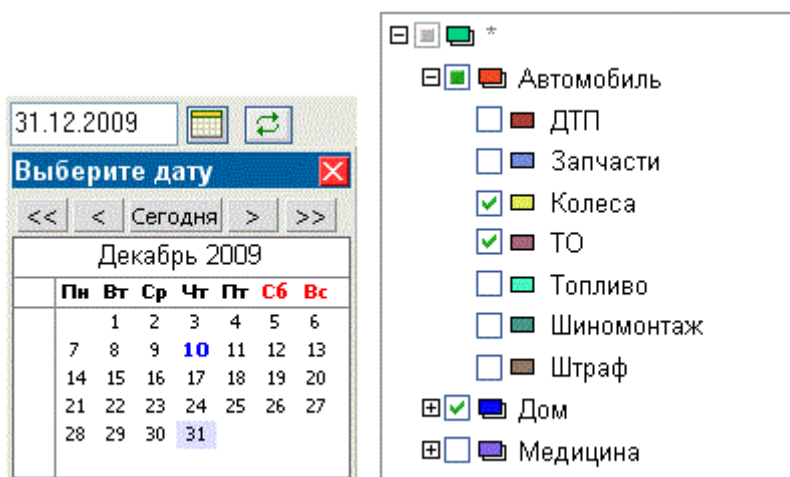


Рис. 5

Средства генерации и разбора XML-документов

В большинстве современных веб-приложений, обмен данными между клиентом (браузером) и веб-сервером осуществляется путем записывания («упаковки») данных в XML-документ и отправки на противоположную сторону, где документ разбирается («parsed») и данные извлекаются для обработки (на серверной стороне) и для отображения результатов (на стороне клиента).

Средства поддержки XML реализованы в виде отдельного статического класса XmlTool, который, в том числе, используется классом AjaxAction на этапах формирования AJAX-запроса и разбора AJAX-отклика (на стороне клиента-браузера).

Средства поддержки арифметики дат

Стандартный класс JavaScript Date поддерживает лишь самые простые операции работы с датами. Для реализации арифметики дат, таких как сложение, вычитание, перевод из григорианской даты в абсолютное значение (юлианский день) и обратно, вычисление дня недели в году и номера недели в году, перевод в строковое значение и обратно (см. тж. следующий раздел), в библиотеку включен класс DateTool.

Расширенная обработка строк

Стандартный класс JavaScript String обладает достаточно развитым функционалом, при этом не поддерживает некоторые удобные методы, включенные в другие платформы, такие как .NET Framework и JDK, а именно: расширенные проверки вхождения (например, endsWith(), startsWith(), contains()) расширенная конвертация в численное значение, конвертация в дату и обратно, средства форматирования и т. д.

В библиотеке TauCode класс String расширен методами, реализующими перечисленные недостающие функции, посредством введения дополнительного класса TauCode.StringTool, а также расширением прототипа класса String дополнительными методами. Например, метод TauCode.StringTool.format() позволяет создавать форматированные строки, как это сделано в классе .NET String.Format().

Валидация данных

Библиотека TauCode включает в себя класс Validator, который позволяет производить проверку корректности введенных пользователем данных до отправки их на сервер, а именно:

- проверка корректности введенных числовых данных (целых или дробных);
- проверка корректности дат;
- проверка корректности строк (с помощью регулярных выражений).

Поддержка бизнес-логики

Библиотека TauCode содержит ряд классов для поддержки бизнес-логики на стороне клиента, в частности:

- классов-описателей сущностей (базовым классом для которых является класс BizEntity);
- класс BizEntityManager, который отвечает за сериализацию/десериализацию бизнес-объектов при отправке данных на сервер/получении данных с сервера, а также служит для создания бизнес-объектов на стороне клиента;
- класс Model, реализующий взаимодействие классов-описателей бизнес-сущностей между собой, в том числе поддержку коллекций бизнес-объектов;
- специализированные классы компонентов пользовательского интерфейса, способные отображать бизнес-объекты, например, EntityGrid (для отображения объектов в таблице), EntityTree (для отображения объектов в дереве).

Обработка исключений

При возникновении исключительных ситуаций (в частности, ошибок) на сервере, встает необходимость корректно известить об этом клиента. В библиотеке TauCode эта задача решена путем «заворачивания» информации об исключении в элемент XML-документа и отправки документа клиенту. Клиент (браузер) при получении такого уведомления будет в состоянии оповестить пользователя о деталях исключения. Данный механизм в библиотеке TauCode реализует класс AjaxException.

Средства отладки и трассирования

Стандартная реализация JavaScript не включает в себя средств отладки и трассирования, предоставляя разработчику лишь метод вывода модального сообщения window.alert(). Библиотека TauCode включает широкие возможности по отладке скриптов, в частности – перетаскиваемое окно трассировки, реализованное посредством DOM. Этот инструмент позволяет отлаживать скрипты без прерывания нормального течения выполнения, что позволяет обнаруживать ошибки, которые сложно диагностировать даже специализированными средствами отладки, такими как FireBug.

Кросс-браузерность

Библиотека TauCode в полной мере отвечает требованиям кросс-браузерности и корректно работает на всех наиболее распространенных браузерах: семейство Gecko (Mozilla Firefox, Netscape Navigator), Internet Explorer, Opera, семейство WebKit (Apple Safari, Google Chrome).

Заключение

Реализованная библиотека TauCode как набор reuse-компонентов для быстрого проектирования (прототипирования) веб-приложений на всех этапах их жизненного цикла практически заложила единую базу построения специализированного инструментария, ориентированного на эффективное решение многих прикладных задач, связанных с обработкой в сетях. Тем самым библиотека TauCode упрощает переход на новые версии программных продуктов в гетерогенных сетевых средах