

Опыт бесшовной интеграции разноязычных Интернет-проектов

П. А. Куратов, Е. Л. Плискин

При разработке новой версии СДО СТ «КУРС» авторам пришлось решать задачу бесшовной интеграции нескольких разноязычных Интернет-проектов. Особенности Интернет-технологий позволяют при решении указанной задачи обходиться без прямого межпрограммного взаимодействия. Для бесшовной интеграции достаточно, чтобы проекты обменивались между собой данными либо через базу данных, либо через дисковые файлы и генерировали выходные документы при помощи общих шаблонов. Неотъемлемой частью решения является генерация кода на различных языках для доступа к хранимым объектам.

1. Введение

1.1. Постановка задачи

В течение полутора лет, начиная со второй половины 2005 г., авторами велась разработка очередной версии Интернет-проекта «Система дистанционного обучения СТ „КУРС“». Предыдущая версия СДО СТ «КУРС» [1] была разработана при участии одного из авторов с применением языков программирования C++ и XML для серверной части СДО и языков DHTML и JavaScript для клиентской части СДО. Здесь и далее под серверной частью программы понимается код, который выполняется под управлением веб-сервера на центральном серверном компьютере СДО, а под клиентской частью понимается код, исполняемый в Интернет-браузере на машине каждого конечного пользователя.

При разработке СДО СТ «КУРС» необходимо было, сохранив функционал старой версии, существенно расширить функционал новой версии СДО с учетом последних стандартов в области дистанционного обучения через Интернет [2], с добавлением инструментов для управления процессами макро-уровня [5] и с использованием порталных технологий. Для ускорения разработки было принято решение дописывать серверную часть СДО на таком языке быстрой разработки, каким является язык PHP.

Функционал, характерный для Интернет-порталов, например, каталог статей об учебных программах, реализован в новой версии СДО СТ

«КУРС» на базе условно-бесплатной издательской системы eZ Publish 2.2.9, версия от 2003 г. [3] Эта система также реализована на языке PHP.

Таким образом, весь корпус кода нынешней версии СДО СТ «КУРС» выпуска 2007 г. (далее «новая версия») складывается из трех разделов:

- Код, унаследованный из базовой версии СТ «КУРС». Языки серверной части: C++ и XML. Языки клиентской части: DHTML и JavaScript. Объем кода около 150 тыс. строк.
- Код, заново разработанный для новой версии СДО. Языки серверной части: PHP и XML. Языки клиентской части: DHTML и JavaScript. Объем кода около 100 тыс. строк.
- Код, унаследованный из издательской системы eZ Publish 2.2.9. Языки серверной части: PHP и HTML. Языки клиентской части: HTML и JavaScript. Объем кода около 150 тыс. строк.

В то время как языки клиентской части указанных разделов кода совпадают, на сервере возникает задача бесшовной интеграции кода на языках C++ и PHP. Заметим, что речь идет об интеграции готовых программ, не рассчитанных на межпрограммное взаимодействие. Требование бесшовной интеграции означает, что конечный пользователь не должен замечать границ между разделами кода. Пожалуй, в любой другой области кроме Интернет-приложений, подобная задача вряд ли могла бы считаться практически разрешимой, даже при наличии исходного кода всех программ. Практичность решения здесь понимается как минимизация трудозатрат разработчиков, что исключает переписывание значительной части старого кода.

Специфика Интернет-приложений, позволяющая ставить и успешно решать задачу бесшовной интеграции разноязычных проектов, следующая:

- Работа Интернет-приложения складывается из выполнения относительно кратковременных запросов на формирование «страниц», наблюдаемых конечным пользователем в браузере.
- Интернет-браузер отображает присланный с сервера код страницы на языках DHTML и JavaScript. Если различные разделы серверного кода будут генерировать сходную структуру кода страницы, то с точки зрения конечного пользователя различные страницы проекта в браузере могут быть внешне похожими, даже если их код сгенерирован на сервере с помощью программ на различных языках программирования.
- Состояние пользователя при работе с системой сохраняется на сервере только в базе данных и в файлах, но не в оперативной памяти сервера.

- Нет необходимости организовывать межпрограммное взаимодействие между составными проектами. Для бесшовной интеграции достаточно, чтобы проекты обменивались между собой данными либо через базу данных, либо через дисковые файлы и генерировали похожие выходные документы.

1.2. Отличительные особенности решения

- Для доступа к хранимым объектам в базе данных на сервере генерируется код на языках C++ и PHP. Для каждой таблицы БД генерируется один исходный файл с определением класса. Объекту сгенерированного класса отвечает одна запись БД. Для каждого поля таблицы БД генерируется член класса. Для загрузки объекта из БД и сохранения изменений в БД генерируются методы класса. Генерация кода выполняется в ходе сборки проекта при каждом изменении исходного файла описания схемы БД на языке SQL.
- Каждый хранимый объект идентифицируется уникальным кодом. Коды объектов отличаются для всех таблиц, а также различаются для объектов, созданных на различных серверах в географически распределенной структуре СДО.
- Генерация страниц в серверном коде СДО на языке C++ производится при помощи XML-шаблонов. В шаблонах используются специальные теги с параметрами, которые позволяют: подставлять данные, извлеченные из БД; включать или исключать фрагменты страницы на основании вычисляемых булевских параметров; повторять подстановку фрагментов с различными параметрами, для отображения списков или таблиц. Шаблоны интерпретируются оригинальным кодом базовой части СДО [1].
- Аутентификация (проверка подлинности) пользователя выполняется сервером один раз в каждом сеансе. Сеанс отвечает одному открытому окну браузера на машине конечного пользователя. Каждому новому сеансу на сервере присваивается уникальный код, под которым информация о сеансе хранится в БД и в файле сеанса. Код сеанса хранится в окне браузера и автоматически посылается на сервер вместе с запросом на получение очередной страницы. Для отправки кода сеанса на сервер в браузере используется механизм "cookie". При помощи этого механизма код сеанса может быть получен исполняемой программой на сервере и использован для извлечения информации о пользователе из БД и текстового файла сеанса.

2. Генерация кода для доступа к хранимым объектам

Подход к ускорению разработки программ, связанный с генерацией кода для доступа к хранимым в базе данных объектам, применялся нами еще в первой версии СДО [1]. В последние годы аналогичный подход развился и получил широкое признание благодаря реализации «типизированного набора данных» (typed dataset) в среде разработки Microsoft Visual Studio 2003/2005 [4]. В этой среде генерация кода осуществляется при помощи графического дизайнера.

В нашем случае при разработке СДО используется консольная прикладная программа, обрабатывающая исходный текст описания таблицы БД на языке SQL для генерации классов на языках C++ и PHP. Среди таблиц БД выделяется таблица `tblEvents`, которая содержит общие свойства для различных типов объектов, например: кодовое обозначение имя и полное наименование объекта, даты его создания и изменения. Запись в таблице `tblEvents` создается одновременно с записью в других таблицах БД, с таким же кодом записи. Например: при создании объекта «учебная программа» создается запись в таблице `tblProgDefs` (определения учебных программ) и запись в таблице `tblEvents` с таким же кодом.

При генерации кода имя класса образуется из имени таблицы БД. Для таблицы `<T>` генерируется класс `CSD0_<T>` на языке C++ и класс `sdo<T>` на языке PHP. Например: для таблицы `tblProgDefs` генерируется класс `CSD0_ProgDef` на языке C++ и класс `sdoProgDef` на языке PHP. В генерированном коде базовым классом для всех других классов является класс, отвечающий таблице `tblEvents`.

Для работы со сгенерированными классами на языке C++ в среде разработки СДО применяются удобные шаблонные классы `SD0_Viewer<T>` и `SD0_Manipulator<T>`. Конструктор указанных шаблонных классов содержит в себе загрузку из БД объекта с заданным кодом. Деструктор шаблонного класса `SD0_Manipulator<T>` содержит сохранение объекта в БД. Изменения автоматически сохраняются в БД при выходе из области действия объектной переменной. Благодаря этим шаблонным классам загрузка объекта из БД и сохранение изменений осуществляется одной строкой кода на C++.

На языке PHP для загрузки хранимого объекта из БД генерируется конструктор вида `ИмяКласса(код объекта)`, а для сохранения изменений в БД в каждом классе генерируется метод `store()`.

В глобальных переменных программы на языках C++ и PHP поддерживается кэш объектов БД, позволяющий повторно обращаться к объекту без лишнего чтения из БД. Использование кэша оправдано благодаря краткости выполнения серверного скрипта в Интернет-

приложении, поскольку для генерации каждой страницы вся программа заново инициализируется.

На базе сгенерированных классов C++ и PHP в ходе разработки СДО создаются производные классы, которые могут обладать дополнительным функционалом и в то же время оставаться хранимыми объектами в БД. Например, для таблицы `tblProgDefs` генерируется класс `sdoProgDef`, а затем вручную создается производный класс `sdoProgDefEx`, в котором реализуются такие методы, как следующие:

- Метод `store()` переопределяет одноименный метод базового класса `sdoProgDef::store()` для сохранения в БД не только самой учебной программы, но и дополнительной информации, включая ссылки на данную учебную программу из других связанных с ней объектов.
- Метод `delete()` для удаления из БД учебной программы вместе со всеми относящимися к ней данными, включая сведения об успеваемости студентов.
- Статический метод `search()` для поиска учебных программ по названию и описанию. Возвращает массив объектов `sdoProgDefEx`.
- Статический метод `selectObject()` для генерации кода на языке DHTML для реализации списка выбора учебной программы, который может отображаться на странице СДО в браузере.

3. Интеграция переменных сеанса между PHP и C++

Основой для интеграции переменных сеанса служит механизм `cookie`, при помощи которого код сеанса становится доступен разделам СДО на языках C++ и PHP. Зная в какой папке сервера размещается файл сеанса, код на языке C++ загружает его и разбирает известный текстовый формат, получая, таким образом, доступ к переменным сеанса PHP. В частности, в переменных сеанса хранятся сведения о пользователе: его учетное имя и список ролей.

В СДО имеется несколько предопределенных ролей пользователей: администраторы, преподаватели, учащиеся и т. п. Каждый пользователь может относиться к одной или нескольким ролям. Список доступных пунктов меню может зависеть от ролей пользователя. Это учитывается при генерации страниц при помощи XML-шаблонов, как объясняется ниже.

4. Генерация страниц при помощи XML-шаблонов

Однородность графического интерфейса страниц СДО, созданных серверными скриптами на языках C++ и PHP обеспечивается цен-

трализованной генерацией оформления страницы. Оформление каждой страницы включает следующие элементы:

- Верхний колонтитул страницы с логотипом и кнопками выбора раздела сайта СДО.
- Левую часть страницы. В ней находится иерархическое навигационное меню и блок регистрации, включающий имя текущего пользователя.
- Нижний колонтитул страницы, в котором может появляться однострочная информация, например уведомление о наличии новых сообщений для пользователя. Немаловажной функцией нижнего колонтитула является также размещение кода на JavaScript, который может зависеть от вычислений в ходе генерации основной части страницы.

Оставшаяся основная часть страницы, наибольшая по площади, генерируется каждым скриптом отдельно.

Оформление страницы генерируется кодом на PHP в виде двух файлов: в первом содержится верхний колонтитул и навигационное меню, а во втором содержится нижний колонтитул. Имя первого файла оформления включает кодовое обозначение выбранного пункта навигационного меню и список ролей пользователя. Это позволяет отображать различное раскрытие навигационного меню и скрывать и показывать пункты меню в зависимости от ролей пользователя. Например, если пункт «ИмяПункта» навигационного меню раздела сайта «ИмяРаздела» выбран пользователем, который принадлежит к ролям «Роль1» и «Роль2», то имя файла для верхнего колонтитула может быть таким:

```
<ИмяРаздела><ИмяПункта>_for_<Роль1>_<Роль1>
```

Нижний колонтитул генерируется одинаково для всех ролей пользователей и не зависит от выбора пункта навигационного меню.

Отдельно генерируется файл, содержащий имя текущего пользователя и кнопки для смены пароля и выхода из системы. Имя этого файла зависит от учетного имени пользователя.

Файлы оформления генерируются кодом на языке PHP и кэшируются в выделенной папке файловой системы сервера, а код на C++ загружает эти файлы и подставляет в шаблон страницы при помощи специального тега `insertHTML`.

При первом входе пользователя в каждый раздел СДО генерируются файлы оформления для всех пунктов меню данного раздела, с учетом списка ролей, которыми обладает текущий пользователь. Поскольку начальная страница каждого раздела СДО создаются кодом на языке PHP, то каждый конечный пользователь обязательно проходит через

генерацию файлов обрамления страниц на языке PHP, прежде чем попадет на страницу, которая генерируется на языке C++.

5. Уникальный код хранимого объекта

Уникальный код хранимого объекта позволяет сопоставить данные, обрабатываемые на различных языках программирования.

Первичным ключом каждой таблицы в БД СДО является код объекта. Код хранимого объекта СДО включает в себя следующие части:

Номер сервера, на котором создан объект. Каждому серверу отведен свой диапазон кодов объектов. Номер сервера занимает старшие разряды кода объекта.

Порядковый номер записи в заданной таблице.

Тип объекта — номер таблицы базы данных от 1 до 99, хранится в младших 2-х разрядах кода объекта.

Наличие уникального кода для различных таблиц БД позволяет использовать общий кэш объектов, а также распознавать по коду тип объекта.

Привязка кода объекта к серверу помогает при репликации правильно удалять из БД все старые объекты, полученные ранее с заданного сервера.

Литература

1. Порай Д. С., Плискин Е. Л., Солощенко Д. В., Голоусикова А. А., Карпенко С. М., Логинов А. С. Опыт разработки Интернет-приложения «Система дистанционного обучения» // Сборник трудов ИСА РАН «Методы и средства работы с документами» / Под ред. Арлазарова В. Л. и Емельянова Н. Е. М.: УРСС, 2000.
2. Спецификация набора стандартов в области дистанционного обучения. SCORM 2004. 3rd Edition: <http://www.adlnet.gov/scorm/index.aspx>
3. Документация по системе eZ Publish 2.2.x: <http://doc22.ez.no/>
4. Visual Basic и основные понятия Visual C#. Введение в наборы данных: <http://msdn.microsoft.com/library/rus/default.asp?url=/library/rus/vbcon/html/vbcondatasets.asp>
5. Куратов П. А., Плискин Е. Л. Инструменты для управления макропроцессами в СДО СТ «КУРС» // В настоящем сборнике.