

УДК 002.6

К.В. Воробьев

Технология разработки информационных систем, использующая принцип динамической загрузки модулей*

С 2001 по 2005 г. автор принимал участие и самостоятельно разрабатывал отдельные подсистемы и блоки различных информационных систем (далее – ИС) делопроизводства и управления в АлтГУ, а также принимал участие в разработках ИС по заказу государственных и коммерческих структур. Была полностью разработана система автоматизации приемной комиссии АлтГУ – информационно-аналитическая система «Абитуриент» (далее – ИАС «Абитуриент»).

Как показал опыт разработки, системы такого рода находятся в состоянии постоянной существенной модификации. Причинами являются:

- постоянно меняющиеся законодательные и нормативные акты;
- быстрая смена социально-экономических условий;
- часто меняющиеся требования заказчика.

Вследствие большого числа модификаций система становится весьма громоздкой, накапливаются ошибки, связанные с отклонениями от начальных условий проектирования.

В связи с этим была предпринята попытка пересмотреть типовые требования к системам такого рода, выделив из них наиболее существенные, применительно к данным проблемам:

1. Расширяемость приложения для возможности последующей доработки.

Исходя из динамичности и непрерывности разработки, необходим механизм расширения и доработки уже сданной в эксплуатацию ИС. Нередко подобная проблема решается выпуском новой версии и обновлением ИС на рабочих местах, но при значительном пространственном распределении рабочих мест представляется трудновыполнимым или невозможным проводить данную операцию достаточно часто.

2. Возможность привлечения команды разработчиков с минимизацией времени стыковки отдельных блоков системы.

Для разработки и поддержки таких систем привлекается достаточно большая команда, состоящая из разработчиков различной специализации. Поэтому становится актуальным вопрос эффективного распределения рабочей силы

в команде. При параллельной разработке отдельных блоков системы большое время, соизмеримое со временем разработки, тратится на стыковку этих блоков. Обычно этот вопрос решается с помощью принятия соглашения о принципах разработки, которые могут быть не соблюдены отдельными членами команды. Однако более эффективным будет создание готовых интерфейсов взаимодействия для программных блоков системы.

3. Возможность оперативной корректировки выполняемых функций ИС.

Для критичных к времени простоя систем, а также на этапе внедрения и опытной эксплуатации необходим механизм оперативной корректировки функциональных возможностей системы и исправления ошибок. Эта задача решается путем прекращения работы и загрузкой новой версии на каждом рабочем месте, что достаточно трудоемко.

4. Разграничение уровня доступа к данным системы и ее функциональным возможностям.

5. Публикация данных в различных форматах.

Важной частью ИС является задача публикации данных, зачастую в различных форматах. Обычно данный модуль выполнен в виде готового набора отчетов или возможности использовать подпрограмму-конструктор для создания отчетов на основе заранее определенных наборов данных. Такие подходы являются недостаточно функциональными, поэтому на основе существующих технологий описания шаблонов требуется создать подсистему публикации произвольных данных системы в различных формах и форматах.

Вследствие этого автором была разработана унифицированная архитектурная схема, удовлетворяющая требованиям 1–5 и не имеющая вышеперечисленных недостатков: клиентское приложение базируется на наборе функциональных единиц (далее – модулей), хранящихся на стороне сервера. В качестве сервера может быть использована СУБД, поддерживающая хранение бинарных данных.

* Данная работа была представлена на Всероссийский конкурсный отбор инновационных проектов аспирантов и студентов по приоритетному направлению «Информационно-телекоммуникационные системы» и была награждена дипломом I степени.



Рис. 1. Принцип работы приложения

Клиентская часть системы представляет собой приложение, устанавливающее соединение с сервером и запускающее стартовый модуль (в рассматриваемой ИАС «Абитуриент» – модуль авторизации). Далее функциональные возможности системы строятся на основе межмодульного взаимодействия и возможностей, вложенных в модули системы. При изменении разработчиком на сервере какого-либо модуля происходит автоматическая актуализация этого модуля на всех клиентских рабочих местах.

Клиентская часть, необходимый программный инструмент и функциональные части модулей могут быть реализованы на одном из современных языков программирования.

Модуль представляет собой совокупность двух файлов:

- декларативного блока, состоящего из описания потоков данных внутри модуля и описания пользовательского интерфейса;
- функционального блока – механизма реакции на события, инициированные пользователем.

Декларативный блок представляет собой XML-поток, состоящий из следующих частей:

1. Описания входных потоков данных из БД, формирующихся на основе параметров, передаваемых модулю, результатом работы является XML-поток данных.

2. Описания пользовательского интерфейса на основе разработанных визуальных компонентов, способных функционировать на основе данных, полученных в п. 1. Также в этой части могут присутствовать инструкции для включения дру-

гих модулей системы, что позволяет выносить повторяющийся элемент продукта в отдельный модуль и вставлять его в другие модули.

3. Описания выходных потоков данных, формирующихся на основе параметров, передаваемых модулю, а также событий, полученных от пользователя. Результатом работы является XML-поток.

4. Описания XSL-шаблонов для формирования различных отчетов на основе данных, полученных в п. 3.

Функциональный блок может быть реализован на высокоуровневом языке программирования в виде подпрограмм-обработчиков событий, полученных от пользователя, причем разработчику доступен весь необходимый инструмент для доступа к компонентам модуля. Совокупность подпрограмм оформляется в виде DLL (семейство ОС Windows) и/или Shared Library (семейство ОС Unix).

Модуль содержит следующие основные подсистемы:

- межмодульного взаимодействия;
- безопасности;
- кэширования;
- публикации данных (подготовка различных отчетов и печать).

Подсистема межмодульного взаимодействия состоит из программного инструментария для вызовов модулей, передачи им параметров, а также для доступа к уже работающим модулям.

Подсистема безопасности представляет собой набор структур данных для хранения учетных

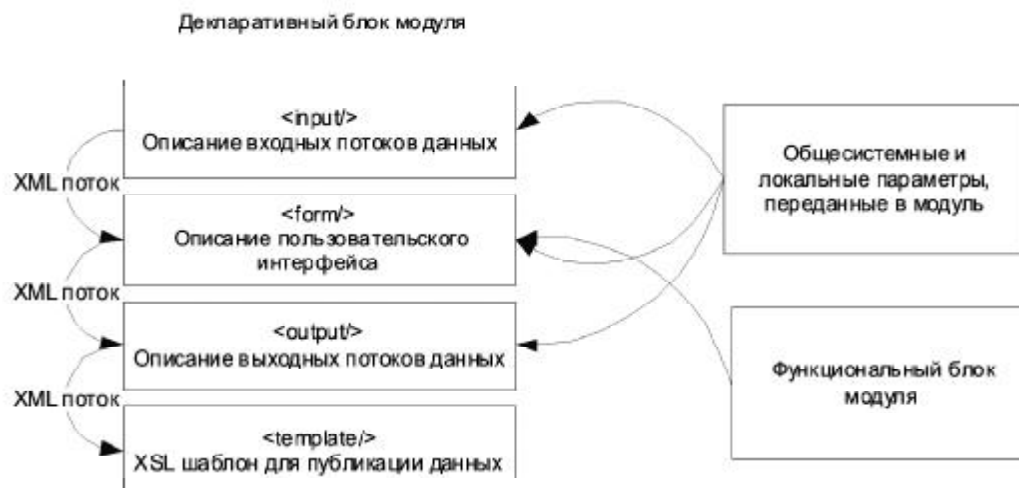


Рис. 2. Схема взаимодействия декларативного и функционального блоков модуля

записей пользователей, организации их в группы. Реализованы унифицированный модуль для авторизации пользователей и механизм ограничения прав доступа как к отдельным пользователям, так и группам, а также к отдельным компонентам модуля и к модулям целиком. Описание прав доступа осуществлено в декларативном блоке модуля.

Подсистема кеширования обеспечивает актуализацию используемых модулей на стороне клиента и позволяет значительно снизить сетевой трафик.

С помощью подсистемы публикации данных на основе XSL-шаблонов можно формировать произвольные отчеты в следующих форматах:

- Microsoft Excel/OpenOffice.org Calc (на основе HTML);
- PDF (на основе технологии FOP);
- текстовые документы для печати на матричных принтерах, с возможностью использования набора тегов для рисования таблиц, идентичных HTML;
- документы с использованием графических примитивов для рисования и вывода текста.

Благодаря тому, что шаблоны являются частью модуля, команда поддержки может оперативно реагировать на изменения различных нормативных документов.

Разрабатываемая по данной технологии ИС начинает полноценно функционировать после регистрации на сервере всего двух модулей – авторизации и управления модулями. После этого система готова для внедрения на рабочие места, выделенные под тестовую эксплуатацию, после внедрения на рабочих местах изменений больше не требуется. Столь ранний выпуск минимальной рабочей версии приложения позволяет разработчикам постепенно уточнять пользо-

вательские интерфейсы и функциональные требования, начиная с первых шагов разработки.

Указанная технология реализована при разработке последней версии ИАС «Абитуриент». В качестве СУБД использовался Oracle Database Server, в качестве языка программирования – Borland Delphi, в качестве клиентской платформы – ОС семейства Microsoft Windows.

В настоящее время ИАС «Абитуриент» прошла промышленную эксплуатацию в период вступительной сессии 2005 г., в течение которой была осуществлена неоднократная и существенная модификация системы. Также данная технология использовалась при разработке коммерческого проекта «Система оперативного учета продаж авиабилетов» для агентства «Авиафлот».

В качестве экономических аспектов использования данной технологии можно выделить следующее:

- сокращение затрат на поддержку и модификацию системы;
- уменьшение времени отклика технической поддержки;
- возможно уменьшение исходного кода системы и сокращение сроков разработки.

Значимость разработки состоит в тиражируемости, либо в плане применения подхода, либо в плане использования готового ядра системы.

В качестве дальнейшего развития данной технологии планируется:

- увеличение числа поддерживаемых программных и аппаратных платформ, СУБД, сетевых протоколов;
- разработка механизма версионности модулей;
- создание банка готовых программных решений для построения на их основе различных ERP-систем.

В результате проведенной работы создана технология разработки и поддержки сетевых информационных систем. Технология удовлетворяет поставленным требованиям, имеет готовую программную реализацию, несколько успешных внедрений. По совокупности достигнутых результатов аналогов данной технологии не выявлено.

Стоит заметить, что перечисленные требования предъявляются к широкому кругу ИС различной степени сложности и профильной направленности. Следовательно, представляется возможным применять данную технологию для создания и поддержки современных ИС.

Литература

1. Брукс Ф. Мифический человек-месяц, или Как создаются программные системы. СПб., 2001.
2. Лингер Р. Теория и практика структурного программирования / Р. Лингер, Х. Миллс, Б. Уитт. М., 1982.
3. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения: Учебный курс MCSD. М., 2000.
4. Иванова Г.С. Технология программирования. М., 2002.
5. Жоголев Е.А. Технология программирования. М., 2004.