ЭЛЕКТРОННЫЕ РЕСУРСЫ.
ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ

УДК 37:004

Д. А. Королёв, А. А. Соболевский, А. С. Сергеев

Изучение возможности адаптации электронных образовательных ресурсов к мобильным платформам

Рассмотрена подготовка комплекта типовых технологических решений и предложений по обеспечению надёжного функционирования электронных образовательных ресурсов нового поколения на мобильных устройствах с платформами Android, Windows, iOS.

Ключевые слова: электронные образовательные ресурсы, электронные учебные модули, мобильные устройства, ОМС-плеер, программа-реализатор, унифицированный пользовательский интерфейс.

Архитектура ОМС-плеера

В последнее время получили распространение открытые образовательные модульные мультимедиа-системы (ОМС), объединяющие электронные учебные модули (ЭУМ). Для воспроизведения модуля на компьютере требуется предварительно установить ОМС-плеер.

ОМС-плеер представляет собой программный комплекс, основное назначение которого – воспроизведение на рабочем месте конечного пользователя электронных образовательных ресурсов нового поколения (ЭОР НП), т.е. ЭУМ.

В процессе воспроизведения ЭУМ ОМС-плеер осуществляет вывод интерактивного аудиовизуального контента в соответствии со сценарием. Сценарий воспроизведения ЭУМ определяется скриптом (JavaScript + XML), размещённым в модуле. В процессе выполнения управляющего скрипта производятся декодирование мультимедиа-компонентов, вывод графических примитивов на экран, воспроизведение звуковых объектов и обработка пользовательских событий.

ОМС-плеер состоит из следующих компонентов: программа-реализатор на базе кроссплатформенного программного ядра; система сопряжения с операционной системой; унифицированный пользовательский интерфейс (УПИ).

Программа-реализатор представляет собой программный компонент, выполненный на языке программирования C++. Основу программы-реализатора составляет кроссплатформенное программное ядро – набор динамически разделяемых библиотек. Кроссплатформенность программного ядра обеспечивается путём компиляции его исходных текстов на C++ под различные целевые платформы.

Кроссплатформенное программное ядро выполняет: интерпретацию программного кода на JavaScript; декодирование мультимедиа-компонентов; синхронизацию аудио- и видеоряда при воспроизведении потокового контента; вывод на экран графических примитивов, звука, сложно форматированного текста; обработку событий пользовательского ввода.

В состав программы-реализатора входит несколько систем. Рассмотрим их.

Система доступа к ресурсам обеспечивает доступ к объектам локальной файловой системы и компонентам для работы с локальным хранилищем. Доступ к любым ресурсам программа-реализатор получает через эту систему, которая допускает подключение модулей расширения, что позволяет программе-реализатору работать с несколькими вариантами локального хранилища.

Система взаимодействия с пользователем принимает и сопоставляет пользовательские события (операции с помощью мыши, клавиатуры) с соответствующими элементами контента и передаёт сведения о входных воздействиях в систему интерпретации сценария для дальнейшей обработки.

Система интерпретации скрипта предназначена для декодирования и интерпретации сценария построения ЭУМ, описываемого на языках *XML* и *JavaScript*. Скриптовый язык позволяет описывать структуру образовательного контента, включающего сцены, состоящие из элементов мультимедиа и их композиций, а также задавать в этих сценах интерактивность.

В качестве декларативной основы языка используется *XML* со специальным набором элементов, их атрибутов и правил вхождения элементов друг в друга. В качестве динамической составляющей используется язык *JavaScript*.

Система декодирования мультимедиа-компонентов распознаёт и декодирует мультимедиа-компоненты во внутренний формат программы-реализатора.

Декодирование всех мультимедиа-компонентов (за исключением Adobe Flash) производится программными компонентами, входящими в состав системы декодирования. Реализация данного подхода позволит минимизировать программное окружение, необходимое для полноценного воспроизведения ЭУМ.

Система воспроизведения мультимедиа-компонентов является низкоуровневой компонентой и предназначена для вывода 2D/3D-графики, видео, звука и синхронизации воспроизведения потокового контента. Для вывода 2D/3D-графики используется библиотека *Irrlicht*, в которой реализована работа с низкоуровневыми сервисами операционной системы, обеспечивающими вывод графических примитивов. Для операционной системы *Windows* возможна работа как через *DirectX 8*, *DirectX 9*, так и через *OpenGL*. Для *Linux* вывод графики производится через *OpenGL*; вывод звуковых объектов – через кроссплатформенную библиотеку *OpenAL*, которая реализует единый интерфейс к низкоуровневым средствам вывода звука на различных аппаратно-программных платформах.

Система сопряжения с операционной системой представляет интерфейс для доступа к сервисам и примитивам операционной системы, таким как:

1. Создание/завершение процесса («потока управления»).

Многие современные операционные системы предусматривают прямую поддержку параллелизма, и это обстоятельство благоприятно сказывается на производительности.

Целью применения принципа параллелизма (многопоточности) является оптимальное использование системных ресурсов. Параллельная обработка обеспечивает повышение пропускной способности приложений на одно- и многопроцессорных компьютерах. За счёт распараллеливания сложность организации программного продукта может быть снижена. Использование параллелизма особенно продуктивно и оправданно сейчас, когда на массовый потребительский рынок вышли современные процессоры, поддерживающие технологии параллелизма на аппаратном уровне.

2. Протокол сокетов реализует механизм взаимодействия между различными процессами, в том числе работающими на различных компьютерах, посредством сетевых сообщений в локальных и глобальных сетях. Интерфейс сокетов поддерживается всеми современными операционными системами, что позволяет единообразно организовать взаимодействие между процессами, выполняющимися под управлением различных операционных систем.

3. Создание примитивов синхронизации – мьютексов, событий, семафоров – и работа с ними.

Для автоматической блокировки объектов спроектирован и реализован «класс-примесь», который использует механизм блокирования объекта выведенного класса при доступе к нему из нескольких потоков управления.

4. Файловые объекты. Для использования в кроссплатформенном программном ядре были выбраны средства работы с файловой системой, реализованные в кроссплатформенной библиотеке «*boost*».

Система концентрирует все системно-зависимые вызовы в одном месте, что облегчает адаптацию программы-реализатора к различным аппаратным и программным платформам. Сервисами, предоставляемыми этой подсистемой, пользуются все подсистемы программы-реализатора.

Унифицированный пользовательский интерфейс (УПИ) реализует контекстно-независимую часть (общую для всех ЭУМ) графического пользовательского интерфейса ОМС-плеера.

При запуске ОМС-плеера на экране появляется первая – унифицированная – часть графического пользовательского интерфейса. После выбора и загрузки модуля интерфейс дополняется контентно-зависимой составляющей. Таким образом, графический пользовательский интерфейс электронного учебного модуля состоит из двух не пересекающихся частей.

УПИ полностью реализован с использованием технологии *Open Scenario Technology* (OST) и представляет собой электронный модуль ОМС. Это означает, что внешний вид и функциональные возможности УПИ можно дорабатывать, исправлять и добавлять независимо от самого плеера ОМС.

Вынесение УПИ из компилируемых компонентов позволяет существовать различным вариантам его исполнения, предназначенным для разных целевых групп (например, по возрастным категориям или физическим ограничениям).

Помимо внешнего визуального интерфейса, УПИ позволяет унифицировать внешний вид создаваемых ЭУМ благодаря тому, что внешние разработчики могут использовать в своих модулях стандартизованные элементы пользовательского интерфейса. Это в свою очередь даёт разработчикам возможность уменьшить время, затрачиваемое на разработку ЭУМ, унифицировать внешний вид ЭУМ, лучше адаптировать дизайн-эргономические характеристики ЭУМ под различные целевые группы пользователей.

В УПИ реализованы:

вывод сообщений об ошибках, возникающих в процессе воспроизведения ЭУМ;

перестроение размеров окна программы-реализатора в соответствии с размерами загруженного ЭУМ;

доступ пользователя к функциям клиентской части среды ОМС;

унифицированный скроллинг;

унифицированная кнопка;

унифицированный графический элемент «перетаскиваемое окно»;

унифицированный графический элемент «диалоговое окно для вывода сообщений пользователю».

Каждый элемент пользовательского интерфейса разработан с таким расчётом, чтобы его можно было использовать в различных контекстах. Для этого все элементы разработаны как самостоятельные объекты.

ОМС-плеер реализован по модульному принципу и имеет механизм расширения функциональности за счёт подключения дополнительных модулей расширения.

Выводы

1. Для создания окна программного модуля, ответственного как за воспроизведение ЭОР НП, так и за работу с Локальным хранилищем, используется кроссплатформенная библиотека *Qt*. Для взаимодействия с пользователем применяются как кроссплатформенные интерфейсы библиотеки *Qt*, так и системно зависимые решения, подключаемые к сборке через директивы препроцессора. Это позволило почти полностью унифицировать на уровне исходных текстов исполняемый модуль для операционной системы *MS Windows* и свободной операционной системы.

2. Органайзер пользователя реализован с использованием кроссплатформенной библиотеки *Qt*, а также библиотеки *gSoap* для работы с Web-сервисами удалённого хранилища в части навигации по совокупному контенту ЭОР НП и получения (закачки) модулей на компьютер пользователя.

3. Почти весь исходный код ПО ЭОР НП, написанный на языке программирования C/C++ (за исключением библиотеки декодирования потокового контента *Intel® Integrated Performance Primitives*), доступен для изучения и адаптации.

Список источников

1. **Мультимедиа** в образовании: контекст информатизации / А. В. Осин. – Москва : Агентство «Издательский сервис», 2014. – 320 с.

2. **Систематизация** информационных ресурсов для сферы образования: классификация и метаданные / А. И. Башмаков, В. А. Старых. – Москва : «Европейский центр по качеству», 2003. – 384 с.

3. **Открытые** образовательные модульные мультимедиа системы / А. В. Осин. – Москва : Агентство «Издательский сервис», 2010. – 328 с.

4. **Программирование** под Android / Зигард Медникс, Лайрд Дорнин, Блэйк Мик, Масуми Накамура. – «Питер», 2012. – 496 с.

5. **Android 3** для профессионалов. Создание приложений для планшетных компьютеров и смартфонов / Сатия Коматинени, Дэйв Маклин, Саид Хашими. – «Вильямс», 2012 – 1024 с.

6. **Описание** проекта Moonlight / Википедия, свободная энциклопедия, 2012 г. – Режим доступа: http://en.wikipedia.org/wiki/Moonlight_(runtime)

7. **Схема** архитектуры ОС Android / Там же. – Режим доступа: http://en.wikipedia.org/wiki/File:Android-System-Architecture.svg