

Г. С. Чернов – студент кафедры компьютерной математики и программирования

С. В. Щекин (канд. техн. наук, доц.) – научный руководитель

ДЕКЛАРАТИВНЫЕ ЯЗЫКИ ОПИСАНИЯ ИНТЕРФЕЙСОВ

Проектирование графического интерфейса является трудоемкой задачей. Для ее облегчения служат различные инфраструктуры GUI, однако они не всегда хорошо документированы и способны освободить разработчика от необходимости написания большого объема кода, что замедляет процесс создания приложения в целом. Кроме того, использование интегрированных сред разработки (IDE) и других графических средств, поддерживающих технологии создания GUI, часто становится причиной появления плохо читаемого и трудного в поддержке кода. Это, в свою очередь, приводит к смешиванию программных слоев, реализующих бизнес-логику и пользовательский интерфейс, усложняя дальнейшее развитие приложения.

Один из подходов к решению этой проблемы заключается в декларативном описании графического интерфейса. Декларативность означает, что описывается желаемый результат, а не последовательность действий для его достижения. Примером такого подхода является HTML, который служит для описания контента, а не функций, при помощи которых он должен генерироваться. Таким образом, декларативные языки не позволяют полностью контролировать процесс фактического создания элементов интерфейса. Это может показаться недостатком, однако существуют и преимущества, в частности, устранение побочных эффектов, таких как модификация глобального состояния (например, глобальных переменных), а также вызов сторонних функций и методов. Кроме того, использование декларативного языка описания UI помогает отделить интерфейс от остального кода приложения. Подобное разделение имеет ряд крайне важных преимуществ, в частности, оно позволяет разграничить роли разработчиков и, как следствие, сократить затраты на интеграцию бизнес-логики с различными технологиями представления данных в приложении.

Эра языков описания интерфейсов началась в 90-х годах 20 века, с появлением языка HTML. К концу 90-х он обрел огромную популярность, прежде всего за счет своей простоты. Для того, чтобы создать сайт не требовалось особых навыков в программировании и специальных инструментальных средств. Любой желающий мог это сделать и почти каждый попробовал. Однако прародитель HTML язык SGML подразумевал структуризацию документов, а это значительно более глубокая модель нежели простое оформление внешнего вида данных. Изначальная идея упорядоченной структуры распределенных данных вернулась вместе с XML и породила эпоху мета описания абстрактных составляющих веб-ресурсов. На этом фоне четко вырисовывалась задача вынесения разметки UI из программного кода приложений. Кроме того, появилась технология каскадных таблиц стилей (CSS), что открыло путь к созданию настраиваемого под конкретное устройство оформления интерфейсов. Эти обстоятельства явились предпосылками созданию языка UIML (User Interface Markup Language). Первая спецификация UIML была представлена компанией Harmonia в январе 1998 года.

UIML – это концепция, в которой путь данных от приложения до физического устройства отображения данных протекает через абстрактные области логики, интерфейса и представления. Область интерфейса, включает описание структуры, стилей, содержания и поведения элементов. Задача языка UIML эффективно реализовать область интерфейса.

Если взглянуть глубже, то выяснится, что UIML определяет следующее:

- составные элементы пользовательского интерфейса;
- каким образом будут представлены элементы пользовательского интерфейса (визуально/вербально/тактильно);
- какого рода содержание будет использовано в пользовательском интерфейсе (текст, изображения, звуки и т.д.);
- какова будет реакция элементов пользовательского интерфейса на действия пользователя;

- каким образом будет производиться контроль событий пользовательского интерфейса (Java Swing или теги HTML);

- с каким внешним API будет взаимодействовать пользовательский интерфейс.

Декларативный язык XUL является частью среды разработки кросс-платформенных интерфейсов, известной как XPFE. Это полнофункциональный язык разметки, на объекты приложений, такие как окна, метки и кнопки. Язык соответствует стандарту W3C XML 1.0. Приложения, написанные на XUL, также могут использовать HTML, CSS, DOM, Java-script. И главное, XUL пытается разделить представление данных и логику приложений. Делается это посредством следующих абстрактных слоев:

- содержание: объявление окон и элементов пользовательского интерфейса ассоциированных с ними;

- оформление: включение CSS и изображений, определение вида приложения;

- локализация: текст, отображаемый в пределах приложения, распределен по специальным локальным файлам, что обеспечивает переносимость языка;

- теоретически XUL обеспечивает кросс-платформенные интерфейсы.

QML – декларативный язык программирования, основанный на JavaScript, предназначенный для дизайна приложений, делающих основной упор на пользовательский интерфейс. Является частью Qt Quick, среды разработки пользовательского интерфейса, распространяемой вместе с Qt. В основном используется для создания приложений, ориентированных на мобильные устройства с сенсорным управлением.

QML-документ представляет собой дерево элементов. QML элемент, так же, как и элемент Qt, представляет собой совокупность блоков: графических (таких, как rectangle, image) и поведенческих (таких, как state, transition, animation). Эти элементы могут быть объединены, чтобы построить комплексные компоненты, начиная от простых кнопок и ползунков и заканчивая полноценными приложениями, работающими с Интернетом.

QML элементы могут быть дополнены стандартными JavaScript вставками путем встраивания .js файлов. Также они могут быть расширены C++ компонентами через Qt framework.

XAML (Extensible Application Markup Language – расширяемый язык разметки приложений) представляет собой язык разметки, используемый для создания экземпляров объектов .NET. Хотя язык XAML – это технология, которая может быть применима ко многим различным предметным областям, его главное назначение – конструирование пользовательских интерфейсов WPF. Другими словами, документы XAML определяют расположение панелей, кнопок и прочих элементов управления, составляющих окна в приложении WPF.

Для работы с XAML используются инструменты разработки, если с XAML работает программист – Visual Studio, если это дизайнер – Expression Blend.

Поскольку оба инструмента поддерживают XAML, можно создать базовый пользовательский интерфейс в Visual Studio, а затем передать его команде дизайнеров, которые доведут его до совершенства, добавив специальную графику с помощью Expression Blend. Фактически такая способность интегрировать рабочий поток разработчиков и дизайнеров – одна из ключевых причин создания Microsoft языка XAML.

Разработчики давно поняли, что создавать сложные, графически насыщенные приложения намного легче, если отделить графическую часть от лежащего в основе кода. Таким образом, художники могут заниматься графикой, а разработчики – кодом. Обе части могут проектироваться и совершенствоваться по отдельности, без проблем, связанных с множеством версий.

Особняком стоит язык Java, являясь самым популярным и часто используемым, он до сих пор не имеет средств для декларативного описания интерфейсов. Визуальная часть приложения зачастую программируется с помощью библиотеки Swing, что влечет за собой вышеописанные проблемы. Разработка средств, для конструирования интерфейсов, на основе декларативного языка, позволила бы увеличить качество и скорость разработки приложения на Java.

Библиографический список

1. Мак-Дональд, Мэтью. WPF 4; Windows Presentation Foundation в .Net 4.0 с примерами на C# 2010 для профессионалов. :Перевод с англ. - М.: ООО «И.Д. Вильямс», 2011. 1024 с.

2. Джасмин Бланшет. Разработка графического интерфейса с помощью библиотеки Qt3.: Prentice Hall PTR, 2010. 304с.
3. QML // Википедия [сайт]. URL:<http://ru.wikipedia.org/wiki/QML>.
4. Языки описания пользовательских интерфейсов // [сайт]. URL: <http://www.raleigh.ru/a/pub/2005/ui-langs.html>.