

УДК 681.3.06

## СРЕДСТВА РАБОТЫ С ПАРАМЕТРИЗОВАННЫМИ ГРАФИЧЕСКИМИ ИЗОБРАЖЕНИЯМИ В ИНСТРУМЕНТАЛЬНОМ КОМПЛЕКСЕ АДЕПТ

О. Б. Арушанян<sup>1</sup>, Н. А. Богомолов<sup>1</sup>, А. Д. Ковалев<sup>1</sup>

Рассматриваются средства работы с параметризованными графическими изображениями, которые используются в рамках разрабатываемого в НИВЦ МГУ программного инструментального комплекса АДЕПТ, предназначенного для автоматизации создания приложений, ориентированных на интерактивную работу со сложными структурами данных. Обсуждаются концепция разработки, особенности реализации, а также различные аспекты использования параметризованных графических изображений в рамках этого инструментального комплекса. Работа выполнена при поддержке РФФИ (код проекта № 05-07-90328).

**1. Введение.** В последние годы в НИВЦ МГУ разрабатывается программный инструментальный комплекс АДЕПТ, предназначенный для автоматизации создания приложений, ориентированных на интерактивную работу со сложными структурами данных [1]. Одним из применений этого комплекса является создание приложений для постобработки результатов вычислительных расчетов. В таких приложениях большую роль играют задачи визуализации данных, требующие построения изображений на основе исходных неграфических данных. Построение подобных параметризованных изображений осуществляется с использованием шаблонов.

В настоящей статье обсуждается подход к автоматизации построения шаблонов изображений, параметризованных на основе исходных неграфических данных. Рассматриваются общая архитектура и особенности реализации программных средств создания и использования графических шаблонов.

**2. Общая схема организации параметризованных изображений.** Под изображением, параметризованным неграфическими данными, будем понимать объект, который позволяет преобразовать некоторое множество наборов исходных неграфических данных во множество характерных для данного объекта изображений. В дальнейшем будем называть такие объекты шаблонами изображений. В изображениях, создаваемых с помощью шаблона, можно условно выделить постоянную часть, вид которой не зависит от исходных данных, и зависящую от них переменную часть.

Достаточно распространенным способом создания шаблонов изображений является представление шаблона в виде программы на алгоритмическом языке. В этом случае итоговое изображение получается в результате работы программы, отдельные фрагменты которой вычисляют зависящие от исходных данных параметры заложенных в шаблоне графических примитивов и порождают с их помощью элементы результирующего изображения. Подобное описание шаблонов обладает высокой гибкостью задания алгоритмов порождения результирующих изображений из исходных данных.

Однако такой подход к описанию шаблонов изображений обладает и рядом недостатков, связанных, прежде всего, с трудностями их создания и последующего изменения. Во-первых, создатель подобных шаблонов вынужден при подготовке шаблона иметь дело с описанием программы генерации изображения на алгоритмическом языке, которое существенно менее наглядно, чем визуальное представление будущего изображения. Во-вторых, шаблон-программа часто может иметь значительный размер, поскольку должна содержать алгоритм порождения полного изображения, которое, помимо элементов, зависящих от исходных данных, содержит и объемную постоянную часть. Визуальные графические редакторы, автоматизирующие создание шаблонов-программ, практически отсутствуют в силу сложности формализации в структурах данных редактора программного контекста, в котором осуществляется построение изображений. И наконец, создатели таких шаблонов должны обязательно обладать достаточно высокой программистской квалификацией.

Предлагаемый авторами подход к созданию шаблонов изображений состоит в представлении шаблона в виде иерархии объектов, содержащих параметры типовых алгоритмов преобразования неграфических

<sup>1</sup> Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, Ленинские горы, 119991, Москва; e-mail: arush@srcc.msu.ru, nbogom@srcc.msu.ru, kovalev@srcc.msu.ru

данных в элементы изображения. Объекты, из которых строится шаблон изображения, далее называемые компонентами, можно разделить на два класса:

- встроенные в инструментальный комплекс визуальные компоненты, непосредственно используемые для генерации изображений;
- создаваемые пользователем невизуальные компоненты, которые осуществляют роль контейнеров наборов данных, используемых для параметризации шаблона.

Визуальные компоненты обеспечивают генерацию широкого спектра изображений от простых до сложных. Примерами простых изображений являются линия, полигон, окружность, квадрат, строка моношрифтового текста и фрагмент готового растрового изображения. Примерами сложных изображений могут служить диаграмма (круговая, линейчатая, столбцовая), ось графика, график функции, фрагмент многшрифтового текста, состоящего из нескольких абзацев, и слой объектов тематической карты. Кроме того, визуальные компоненты содержат встроенные алгоритмы автоматического размещения изображений отдельных компонентов относительно друг друга, которые учитывают, в том числе, и иерархию связей компонентов шаблона.

Наборы данных, хранящиеся в невизуальных компонентах, могут содержать как исходные числовые данные, так и явно заданные параметры изображения: строки, наборы цветов, описания шрифтов и др. Эти данные используются для динамической настройки параметров визуальных компонентов шаблона. Такая настройка осуществляется непосредственно в процессе построения результирующего изображения на основе индивидуально заданных в каждом компоненте шаблона указаний на рассылку данных. Если оказывается, что применения средств простой рассылки данных недостаточно, то более сложные взаимосвязи между параметрами шаблона могут быть описаны на встроенном в комплекс АДЕПТ алгоритмическом языке SPL [2].

Иерархическую структуру компонентов шаблона в силу наличия гибких средств динамической настройки параметров компонентов, учитывающих, в том числе, и контекст иерархических связей компонентов шаблона, можно рассматривать также и как программу на специализированном высокоуровневом алгоритмическом языке описания параметризованных изображений.

Невизуальные компоненты (контейнеры данных) являются удобным интерфейсом между шаблоном изображения и использующей его прикладной программой. Благодаря наличию компонентов контейнеров данных прикладная программа может “не знать” деталей организации шаблона при передаче ему параметров для построения очередного изображения. Это позволяет, например, организовать единый интерфейс прикладной программы с различными вариантами шаблонов, имеющими значительные отличия в реальном составе и структуре иерархических связей входящих в его состав визуальных компонентов.

Процесс подготовки и использования шаблонов изображений состоит из следующих этапов:

- подготовка шаблона с помощью визуальных интерактивных редакторов;
- заполнение интерфейсного компонента шаблона необходимыми параметрами;
- запуск интерпретатора шаблона, осуществляющего генерацию требуемого изображения в заданном формате.

Создание шаблона выполняется до начала работы использующей его прикладной программы. Заполнение интерфейсного компонента шаблона параметрами и запуск интерпретатора шаблона осуществляется прикладной программой. Таким образом, для генерации изображений с помощью шаблона прикладная программа должна иметь доступ к интерфейсному компоненту шаблона и к его головному компоненту, необходимому для запуска интерпретатора шаблона.

Результирующий образ, генерируемый интерпретатором шаблона, может представлять собой как растровое изображение (BMP, JPG и т.д.), так и изображение в векторной форме (EMF, VML, SVG и др.).

В предлагаемом авторами подходе к организации шаблонов изображений сочетается достаточная гибкость задания алгоритмов порождения результирующих изображений, характерная для шаблонов, которые представлены в виде программы на алгоритмическом языке, с возможностью использования интерактивных визуальных редакторов, облегчающих процесс подготовки шаблонов и позволяющих привлечь к подготовке шаблонов специалистов, не обладающих программистской квалификацией.

Автоматизация создания шаблонов изображений в рамках предлагаемого подхода достигается за счет следующих особенностей их реализации:

- наличия визуальных компонентов, позволяющих создавать такие сложные изображения, как диаграмма, график функции, слой тематической карты, легенда карты или диаграммы и т.д.;
- наличия средств автоматического размещения изображений отдельных компонентов, учитывающих, в том числе, и их положение в иерархии компонентов шаблона;

- наличия средств визуализации шаблонов, облегчающих процесс их создания и изменения;
- наличия интерактивных проблемно-ориентированных редакторов шаблонов (редактор иерархии компонентов шаблона, редакторы различных групп параметров компонентов и отдельных полей, векторные графические редакторы точечных, линейных и площадных объектов).

**3. Компоненты шаблона изображения.** Каждый компонент шаблона обладает определенным набором параметров, которые используются интерпретатором шаблона в процессе построения изображения. Одни параметры являются общими для компонентов всех типов, некоторые используются лишь для визуальных компонентов любого типа, а часть параметров индивидуальна для каждого типа визуальных компонентов. Параметры компонентов разбиты на следующие группы:

- общие параметры, управляющие “исполнением” компонентов интерпретатором шаблона, включая динамическую настройку компонентов шаблона в процессе создания изображения;
- параметры, управляющие размещением (размером и положением) и являющиеся общими для визуальных компонентов всех типов;
- индивидуальные параметры визуальных компонентов, состав которых определяется типом компонента (например, параметры шрифта для отображения текста или толщина и цвет для отображения слоя линейных объектов тематической карты);

- “пользовательские” параметры для хранения исходных числовых данных и атрибутов отображения, которые применяются для динамической настройки визуальных компонентов шаблона.

Каждый компонент шаблона изображения содержит два набора параметров: статические параметры, задаваемые на этапе предварительной подготовки шаблона с помощью специальных редакторов, и динамические параметры, формируемые интерпретатором шаблона в процессе исполнения компонента и реально используемые для построения результирующего изображения.

Для формирования динамических параметров используется следующий алгоритм. В начале процесса генерации изображения значения статических параметров всех компонентов переносятся в их динамические параметры. При последующем обходе иерархии компонентов шаблона интерпретатор выполняет связанные с каждым компонентом задания на рассылку или преобразование параметров, которые изменяют начальные значения динамических параметров.

**3.1. Общие параметры управления компонентами.** Общие параметры компонентов, управляющие работой интерпретатора шаблона, позволяют влиять на обход иерархии компонентов, включая запрет исполнения отдельных компонентов и прерывание работы интерпретатора. Кроме того, эти параметры позволяют управлять динамической настройкой параметров компонентов прежде всего за счет рассылки данных из пользовательских параметров любых компонентов в параметры компонентов, непосредственно управляющих генерацией изображения визуальных компонентов.

Для управления динамической настройкой в каждом компоненте можно задать индивидуальный список действий по динамической настройке. Каждое действие представляет собой либо указание на пересылку заданного фрагмента данных внутри шаблона, либо вызов процедуры на встроенном языке SPL для некоторой дополнительной обработки данных.

Указание на пересылку заданного фрагмента данных внутри шаблона позволяет задать произвольный источник и получатель данных. В качестве источника и получателя данных можно указать любое поле любого компонента набора. При задании компонента источника и компонента получателя данных можно использовать как абсолютные (явно заданные) значения ссылок на компоненты шаблона, так и относительные (неявно заданные) ссылки: ссылка на “себя”, ссылка на “родительский” компонент, ссылка на компонент, заданный в некотором общем параметре, и др.

**3.2. Параметры управления размещением визуальных компонентов.** Каждый визуальный компонент характеризуется некоторой прямоугольной областью — пространством компонента. Положение и размер изображения, генерируемого компонентом, жестко связано с его пространством. При задании пространства компонента можно независимо задавать положение левой, верхней, правой и нижней его границ. При этом для каждой пары границ вдоль горизонтали (левая/правая) и вдоль вертикали (верхняя/нижняя) можно задавать любые два значения из трех возможных:

- положение первой границы,
- положение второй границы,
- размер пространства компонента вдоль выбранного направления.

При задании положения границ пространства компонента можно использовать в качестве базового либо головной компонент шаблона, либо родительский компонент, либо прямоугольный фрагмент пространства родительского компонента, оставшегося “свободным” после размещения дочерних компонентов этого родительского компонента.

Для задания положения и размера пространства компонента могут быть использованы относительные единицы (доля от размера базового компонента), абсолютные единицы результирующего изображения (миллиметры, пункты, пиксели), либо произвольные пользовательские координаты, чаще всего используемые при описании слоев координатных объектов тематических карт.

Пространство компонента может быть отображено в результирующем изображении; для этого можно задать цвет фона заливки прямоугольной области пространства компонента, а также цвет и толщину ограничивающей эту область рамки.

**3.3. Пользовательские параметры.** Эти параметры представляют собой произвольный набор именованных фрагментов данных. Набор пользовательских параметров может быть задан для любого компонента шаблона. Главное назначение пользовательских параметров состоит в обеспечении прикладных программ удобным интерфейсом по управлению шаблонами изображений.

**4. Редактирование шаблонов изображений.** В инструментальном комплексе АДЕПТ создание и изменение шаблонов изображений осуществляется с помощью ряда проблемно-ориентированных интерактивных редакторов. Основной редактор шаблонов обеспечивает просмотр и редактирование иерархии компонентов шаблона как в виде дерева узлов, так и в виде результирующего изображения. Редактор позволяет просматривать результирующее изображение в различных масштабах, в том числе и в натуральном, когда одному пикселю экрана соответствует пиксель устройства, на которое планируется осуществлять вывод готового изображения, например пиксель принтера.

Основной редактор шаблонов позволяет вызывать специализированные интерактивные редакторы различных перечисленных выше групп параметров выбранных компонентов, которые, в свою очередь, могут вызывать специализированные интерактивные редакторы отдельных полей — редакторы цвета, числового скаляра, координат точки или прямоугольника. Изменения параметров, сделанные специализированными редакторами параметров компонентов, сразу же отображаются в результирующем изображении, что позволяет удобно подбирать параметры шаблона в режиме WYSIWYG.

Кроме того, в режиме WYSIWYG работают и векторные графические редакторы точечных, линейных и площадных объектов, используемые для редактирования координатных объектов тематических карт.

**5. Заключение.** Шаблоны изображений, реализованные в инструментальном комплексе АДЕПТ, были практически применены, в частности, при создании растровых шаблонов [3] для Интернет-атласа [4] по уровню жизни в регионах РФ, разработанного в рамках проекта РФФИ № 05-07-90328.

#### СПИСОК ЛИТЕРАТУРЫ

1. Арушанян О.Б., Богомолов Н.А., Ковалев А.Д., Волченкова Н.И. Об одном подходе к автоматизации создания приложений, ориентированных на работу со сложными структурами данных // Вычислительные методы и программирование. 2005. 6, № 1. 115–123.
2. Арушанян О.Б., Богомолов Н.А., Ковалев А.Д. Встроенный язык для инструментального комплекса, ориентированного на интерактивную работу со сложными структурами данных // Вычислительные методы и программирование. 2006. 7, № 1. 131–143.
3. Арушанян О. Б., Богомолов Н.А., Ковалев А.Д., Сеницын М.Н. Об одном подходе к построению графических изображений в среде Интернет-браузера // Тр. Всероссийской научной конференции “Научный сервис в сети Интернет”. М.: Изд-во Моск. ун-та, 2004. 132–133.
4. Арушанян О.Б., Богомолов Н.А., Ковалев А.Д., Сеницын М.Н. Архитектура клиентского программного обеспечения для WEB-приложений, ориентированных на представление данных // Вычислительные методы и программирование. 2004. 5, № 2. 178–191.

Поступила в редакцию  
10.01.2007