Уфа : УГАТУ, 2010 Вестник УГАТУ Т. 14, № 5 (40). С. 170–175

УДК 004.738.5

В. В. Миронов, К. Э. Маликова

ИНТЕРНЕТ-ПРИЛОЖЕНИЯ НА ОСНОВЕ ВСТРОЕННЫХ ДИНАМИЧЕСКИХ МОДЕЛЕЙ: ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Обсуждается класс интернет-приложений на основе встроенных динамических моделей. Рассматриваются правила построения интернет-страницы, структуры элементов управления пользовательского интерфейса для данного класса приложений, определяются правила интерпретации элементов управления пользовательского интерфейса. *Интернет-приложение*; *динамическая модель*; *встроенная модель*; *элементы управления пользовательского интерфейса*; *XML-технологии*

ВВЕДЕНИЕ

В работах [1, 2] авторами предложен новый класс интернет-приложений – на основе встроенных динамических моделей. На концептуальном уровне обсуждались вопросы построения таких приложений, рассматривались вопросы разработки архитектурной модели данного класса приложений, а также организация взаимодействия ключевых архитектурных компонентов и их функциональность, однако оставались открытыми вопросы структуры и правил построения элементов пользовательского интерфейса. Данная статья является продолжением упомянутых работ в плане определения правил построения и структуры элементов пользовательского интерфейса.

Элементы пользовательского интерфейса, т. е. компоненты, предоставляющие программные интерфейсы, предназначенные для многократного использования на интернет-страницах, являются неотъемлемой частью архитектурной модели любого интернет-приложения [3].

В соответствии с предложенной концепцией построения интернет-приложений на основе встроенных динамических моделей в составе встроенной модели были выделены следующие наиболее часто используемые категории элементов управления: элементы управления данными, обеспечивающие автоматическое создание и передачу пользователю персонализированных документов в необходимом формате в соответствии с переданными пользователем данными; элементы управления навигацией, обеспечивающие инициализацию пользовательского события и автоматическое погружение в соответствующее состояние с последующим добавлением его в модель памяти текущих состояний; элементы управления входом, поддерживающие программную аутентификацию пользователей, подключение к хранилищу данных, нахождение подробной информации о конкретном пользователе, проверку введенных удостоверений; элементы управления интернет-страницей, поддерживающие построение сложного сочетания HTML-элементов в контексте интернет-приложений на основе встроенных динамических моделей.

Непосредственный вывод HTML-представлений пользовательских элементов управления, а также интернет-приложения в целом осуществляется на интернет-страницах. В этой связи обязательной частью архитектуры любого приложения также являются интернет-страницы, программный код которых обрабатывается последовательно от начала до конца. Для каждой интернет-страницы разработчику необходимо обеспечить создание некоторого постоянного объема программного кода – выделить отдельные логические части в соответствии со спецификацией HTML и правилами целевого языка программирования. Таким образом, каждая страница, отвечающая HTML-спецификации, должна иметь определенную структуру: содержать дескрипторы, отмечающие начало и завершение страницы; заголовок; название, отражающее содержимое текущей интернет-страницы; ссылки на правила, описывающие представление внешнего вида и текущую кодировку интернет-страницы. Структура страницы

Контактная информация: (347) 272-89-81

Работа выполнена в рамках научной школы УГАТУ «Теория и практика разработки информационных систем» и поддержана грантом РФФИ 10-07-00167-а «Электронные документы со встроенной динамической моделью», а также грантом Президента Российской Федерации НШ-65497.2010.9 для ведущих научных школ

приложений на основе динамических моделей не является чем-то новым, но процессы ее генерации и настройки на пользовательские предпочтения являются несколько иными.

1. СТРУКТУРА ИНТЕРНЕТ-СТРАНИЦЫ

Формирование HTML-представления страницы интернет-приложений на основе встроенных динамических моделей осуществляется в результате интерпретации динамической модели, размещенной на серверной стороне с учетом доступных состояний и ассоциированных с ними прикладных фрагментов. В этой связи каждому состоянию, входящему в состав динамической модели, соответствует некоторая область (или ее часть) HTML-представления. Данная область представления, в свою очередь, включает постоянную область (дескрипторы, определяющие структурную организацию интернет-страницы), содержимое которой не зависит от текущего состояния, и переменную область, содержание которой определяется только текущим состоянием конкретного пользователя.

Кандидатами на размещение в контексте постоянной области являются дескрипторные элементы, HTML-представления которых размещаются строго в определенных областях страницы каждого интернет-приложения, а также не зависят от текущего пользовательского состояния. В качестве таких элементов можно выделить дескрипторы, обозначающие начальную и завершающую части интернет-страницы, а также тело интернет-формы.

Для исключения дублирования поведения, предполагающего многократное извлечение и последующую интерпретацию дескрипторных элементов, составляющих постоянную область интернет-страницы, целесообразным решением будет являться их вынесение за пределы встроенной динамической модели непосредственно в программный код интерпретатора.

В ходе взаимодействия пользователя с интернет-приложением необходима поддержка интерактивности данного процесса. В этой связи важным «строительным блоком» динамических интернет-приложений является элемент интернет-формы, предназначенный для ввода и передачи данных пользователем, фиксации событий, влияющих на дальнейшее поведение приложения в целом.

Как и в традиционных приложениях, интернет-форма создается с помощью парных дискрипторов (<form>), внутри которых и помещается необходимое представление страницы. В связи с тем, что структура представления интернет-страницы приложений данного класса является оверлейной и состоит из налагаемых друг на друга областей, интернет-форма по своей сути является объектом-контейнером для всего множества элементов, составляющих переменную область – элементов управления пользовательского интерфейса, а также прикладных фрагментов, включающих различные участки HTML-представлений и сценариев на целевом языке программирования, доступных в контексте текущего пользовательского состояния. Таким образом, структура интернет-страницы включает постоянную область, определяющую структурную организацию страницы и интернет-формы, а также переменную область, входящую в состав интернет-формы, содержащую HTML-представление данной страницы в контексте текущего состояния.

Необходимо отметить, что каждый из элементов управления, в свою очередь, может содержать неограниченное количество дочерних пользовательских элементов управления, отвечающих за генерацию собственного программного кода. Некоторые из данных элементов управления являются простыми и отображаются на интернет-странице на определенные HTML-дескрипторы (элементы управления интернет-страницей, навигационные элементы управления), другие же являются более абстрактными и реализуют более сложное представление (элементы управления входом, данными). По мере интерпретации элементов управления интерпретатор добавляет полученное HTML-представление к представлению интернет-страницы, которую отправляет на сторону клиента.

2. СТРУКТУРА ЭЛЕМЕНТОВ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

При первом погружении в динамическую модель, а также в ходе взаимодействия клиента с интернет-приложением на вход интерпретатора поступают текущие значения пользовательских элементов управления, определенных в контексте текущего состояния. В этой связи на первом проходе интерпретатору необходимо не только однозначно определить категорию элемента управления, с помощью которого пользователем было идентифицировано событие-источник, но и однозначно определить событие-цель, а затем выполнить необходимые действия, предписанные данным элементом управления и на втором проходе обеспечить

генерацию HTML-представления данного элемента управления, не нарушая при этом структуру интернет-страницы.

Для решения поставленных задач определим структуру и правила построения элементов пользовательского интерфейса следующих категорий: управления интернет-страницей, управления входом, управления данными, управления навигацией.

Элементы управления интернет-страницей. Для решения задач, связанных с поддержкой построения сочетания HTML-элементов любой сложности, их единообразного представления, а также введения специфических свойств, событий и методов в контексте интернет-приложений на основе встроенных динамических моделей, данный тип элементов управления пользовательского интерфейса был выделен в отдельный класс.

Ввиду того, что набор данных элементов обширен, в качестве примера рассмотрим элемент управления, отвечающий за формирование групп переключателей, концептуальная схема которого приведена на рис. 1.

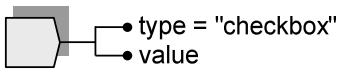

Рис. 1. Концептуальная схема элементов управления интернет-страницей

Как видно из предложенного рисунка, внешний вид и основное поведение данного элемента является унаследованным от HTML-спецификации, ключевыми отличиями является то, что в качестве дополнительных атрибутов были введены: атрибут type, который однозначно идентифицирует тип элемента управления в ходе обработки динамической модели; атрибут value, содержание которого представляет массив, формируемый путем разделения переменных по границам, образованным сепаратором (в рассматриваемом случае в качестве сепаратора используется пробел); ввиду того, что уникальное имя, которое задается непосредственно в дескрипторе пользовательского элемента, однозначно идентифицирует данный элемент управления, атрибут name удален как избыточный.

Несмотря на то, что элементы управления интернет-страницей не имеют расширенных свойств и их создание не является сложным, с их помощью невозможно решить каждую задачу, связанную с управлением.

Например, в рамках рассматриваемого подхода встроенная динамическая модель может содержать персонализованные подмодели, доступные только для аутентифицированных пользователей, также различен механизм отслеживания ее текущих состояний для аутентифицированных и анонимных пользовательских групп. Для поддержки процедур пользовательской аутентификации в динамической модели предусмотрен специальный элемент управления входом в подмодель, соответствующей аутентифицированному пользователю.

Элементы управления входом позволяют реализовать решения управления доступом к персонализованным областям интернет-приложения на основе проверки подлинности пользователей. На рисунке ниже приведена концептуальная схема данного класса элементов управления.

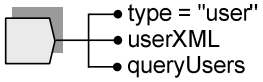

Рис. 2. Концептуальная схема элементов управления входом

В соответствии со структурной схемой данного элемента управления (рис. 2) из множества атрибутов можно выделить следующие:

- type указывает на то, что данный элемент управления представлен в классе элементов управления входом;
- userXML определяет надлежащий источник хранения пользовательских учетных записей (в рассматриваемом случае для данных целей используется соответствующий XML-документ);
- queryUsers определяет аутентификационный запрос доступа к учетной записи пользователя.

Таким образом, предполагается наличие некоторого хранилища аутентификационной информации пользователей. При инициализации события, предполагающего проверку подлинности пользователя, интерпретатором идентифицируется источник хранения пользовательских учетных записей (userXML), осуществляется подключение к данному источнику, а также на основании запроса (queryUsers) доступа выполняется проверка наличия учетной записи пользователя, а затем аутентификационная информация сравнивается с хранящейся на стороне сервера.

Элементы управления данными. Интернет-приложение также должно обеспечивать возможность программной генерации персонализованных документов в необходимом форма-

те в соответствии с переданными пользователем данными.

Данный класс пользовательских элементов управления представлен для обеспечения данной возможности, концептуальная схема элементов управления данными приведена на рис. 3.

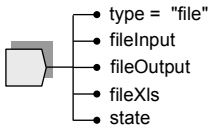

Рис. 3. Концептуальная схема элементов управления данными

В соответствии со структурной схемой элементов управления данной категории, множество атрибутов делится на следующие непересекающиеся подмножества:

- type указывает на то, что данный элемент управления представлен в классе управления данными;
- fileInput идентифицирует исходный файл, содержанием которого является набор пользовательских данных;
- fileOutput определяет имя целевого персонализованного документа, а также необходимый формат его вывода;
- fileXls определяет таблицу стилей для преобразования пользовательских данных в целевой документ;
- state определяет событие, инициализирующее данный элемент управления.

Приведенный ниже пример отображает структурное описание элемента управления данными, предназначенного для формирования персонализованного документа «Явочный лист» на основании пользовательских данных.

```
<file type='file'
      fileOutput='Сопровод.документы.doc'
      fileInput='Электронное дело'
      fileXls =' Явочный_лист.xls'
      state ='Сформировать'>
 </file>
```

Необходимо отметить, что структура данного элемента управления допускает только ограниченное взаимодействие, которое можно расширить путем введения в значения атрибутов участков кода, программно определяющих имя целевого персонализованного документа, а также документов, отвечающих за преобразования пользовательских данных в целевой документ. В этом случае значения данных атрибутов вступят в зависимость от значений других элементов управления (например, управления интернет-страницей).

Элементы управления навигацией. Последним и наиболее часто используемым классом элементов управления пользовательского интерфейса являются элементы управления навигацией, обеспечивающие основу взаимодействия между пользователем и интернет-приложением – инициализацию пользовательского события, автоматическое погружение в соответствующее состояние и последующую коррекцию модели памяти текущих состояний. На основании предложенной концепции в динамическую модель добавлен специальный элемент jumpbutton, предоставляющий единообразный способ конфигурации кнопочного элемента управления. Концептуальная схема элемента управления навигацией представлена на рис. 4.

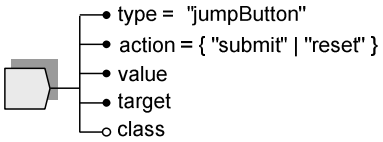

Рис. 4. Концептуальная схема элементов управления навигацией

Как и в традиционных интернет-приложениях, структура навигационных элементов управления включает атрибуты:

- value определяет текстовое отображение кнопочного элемента управления на странице;
- action определяет действия, которые необходимо выполнить после инициализации события данным элементом управления – отправить данные на сторону сервера или вернуться к их первоначальному значению на стороне клиента;
- class предоставляет именованный набор атрибутов, который применяется к элементу управления для назначения стилевого оформления. Значения атрибутов не влияют на общую функциональность интернет-приложения, ввиду этого наличие значений атрибутов является необязательным.

Особенность данного класса интернет-приложений состоит в том, что его работа управляется встроенной динамической моделью на основании отслеживания экземпляров памяти текущих состояний. В этой связи необходимо четко определять состояние-цель, а также проход, на котором выполняется обработка данного элемента управления и генерация его HTML-представления. Указанные особенности предо-

пределяют необходимость введения дополнительных атрибутов в структуру пользовательских элементов управления навигацией:

- type указывает на то, что данный элемент управления представлен в классе управления навигацией;
- target определяет состояние-цель, в которое необходимо выполнить погружение при инициализации события при помощи данного элемента управления пользователем.

Для однозначной идентификации конкретного навигационного элемента управления из всего множества элементов управления навигацией используется уникальное имя, которое задается непосредственно в дескрипторе пользовательского элемента управления. В этой связи атрибут name был удален как избыточный. Приведенный ниже пример отображает структурное описание навигационного элемента управления (jumpbutton) – «Телефон», предназначенного для перехода в состояние отображения контактного телефона из текущего состояния отображения электронного адреса пользователя:

```
<J-Email type='jumpButton' action="submit"
        target = "Телефон"
        value="Дать Телефон"
        class="styleJumpButton"/>
</ J-Email>
```

Для генерации на основе элементов управления соответствующего HTML-представления интернет-страницы следует однозначно выполнить преобразования значений атрибутов элементов управления в свойства соответствующих HTML-дескрипторов. В случае же инициализации (на следующем шаге сеанса) определенного события данным элементом выполнить закрепленные за элементом управления действия. В этой связи необходимы собственные правила интерпретации, которые расширят единый алгоритм интерпретации встроенной динамической модели.

3. ИНТЕРПРЕТАЦИЯ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

В ходе интерпретации встроенной динамической модели рекурсивно совершается полный обход дерева модели в контексте текущих состояний и иерархически генерируется структура интернет-страницы. На начальном этапе погружения генерируется структурная заготовка интернет-страницы, включающая начальную часть (<html>, (<head>, <title>, <body>), а также различные дополнительные элементы, определенные разработчиком приложения.

Например, для представления образующих внешний вид каждой интернет-страницы элементов управления используются правила технологии CSS[1], включающие их описательное представление. Следующим шагом формирования интернет-страницы является процесс построения интернет-формы (<form>), в тело которой в ходе дальнейшей интерпретации встроенной модели помещаются различные HTML-представления прикладных фрагментов и элементов управления. На заключительном этапе (при погружении в конечное состояние встроенной динамической модели) интерпретатор завершает работу построением завершающей части интернет-формы (</form>), а затем и интернет-страницы (</body>, </title>, </head>, </html>), в результате формируется соответствующая текущему состоянию интернет-страница.

Рассмотрим более подробно процесс формирования содержания интернет-формы. В ходе интерпретации динамической модели каждое состояние модели (stateHSM) проверяется на наличие дочерних элементов (xHSM), представляющих конгломерат субмоделей, переходов, а также элементов управления, принадлежащих различным классам. Если в контексте текущих состояний было определено наличие элементов управления пользовательского интерфейса, то осуществляется генерация их HTML-представлений. Данный процесс предполагает интерпретацию значений атрибутов элементов управления и их преобразование в соответствующие свойства HTML-дескрипторов, определяющих представление внешнего вида данных элементов управления на интернет-странице.

В качестве примера ниже приведен процесс преобразования атрибутов навигационного элемента управления (jumpButton) в соответствующие свойства HTML-дескриптора (button) (в рассматриваемом случае для данных целей используется конструкция языка PHP – echo):

```
echo '<input type="'.$xHSM -> getAttribute ("type").'"
      class ="'.$xHSM -> getAttribute ("class").'"
      value="'.$xHSM -> getAttribute ("value").'"
          name="'.$xHSM -> tagName.'"  />';
```

В случае инициализации определенного события в ходе взаимодействия пользователя

[1] CSS (Cascading Style Sheets – каскадные таблицы стилей) – технология описания внешнего вида документа, написанного на языке разметки.

с интернет-приложением при помощи элемента управления (нажатие кнопки, выбор элемента в списке) осуществляется выполнение одного из следующих вариантов действий: 1) если инициализация данного события не предполагает переход в новое текущее состояние, то осуществляется фиксация уникального имени данного элемента управления в глобальный массив интернет-приложения; 2) если инициализация данного события предполагает переход в новое текущее состояние, то на первом проходе (FirstPass) выполняется коррекция модели текущих состояний на основании состояния-цели, определенного элементом управления. Для выполнения действий данного характера предназначена специальная обрабатывающая функция, которая обеспечивает добавление нового текущего состояния (если модель является темпоральной) или замену предыдущего текущего состояния новым (в режиме «работа без предыстории»). На втором проходе (SecondPass) интерпретатором осуществляется формирование новой интернет-страницы, ассоциированной с новым текущими состоянием динамической модели.

ЗАКЛЮЧЕНИЕ

В статье обсуждается новый класс приложений на основе встроенных динамических моделей, характеризующийся следующими особенностями построения HTML-представлений интернет-страниц:

- структура интернет-страницы включает постоянную область, содержимое которой не зависит от текущего состояния, и переменную область, содержание которой определяется только текущим состоянием конкретного пользователя;
- в контексте постоянной области размещаются дескрипторные элементы, обозначающие начальную, завершающую части интернет-страницы, а также структуру интернет-формы.
- элементы управления данными, навигацией, входом, а также интернет-страницей входят в состав переменной области;
- согласно структуре встроенной модели состояние может включать элементы управления пользовательского интерфейса, в этой связи каждое состояние встроенной модели проверяется на наличие соответствующих элементов управления;
- в случае инициализации определенного события пользователем при помощи элемента управления, не предполагающего переход в новое текущее состояние, осуществляется фиксация уникального имени данного элемента управления в глобальный массив;
- в случае инициализации определенного события пользователем при помощи элемента управления, предполагающего переход в новое текущее состояние, осуществляется коррекция модели текущих состояний и формирование интернет-страницы в контексте нового текущего состояния.

СПИСОК ЛИТЕРАТУРЫ

1. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: идея, концепция, безопасность // Вестник УГАТУ: научн. журн. Уфимск. гос. авиац. техн. ун-та. 2006. Т. 13, № 2. С. 167–179.

2. **Миронов В. В., Маликова К. Э.** Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация // Вестник УГАТУ: научн. журн. Уфимск. гос. авиац. техн. ун-та. 2010. Т. 14, № 2. С. 154–163.

3. **Купер А., Рейман Р., Кронин Д.** Алан Купер об интерфейсе. Основы проектирования взаимодействия. СПб.: Символ-Плюс, 2010. 688 с.

ОБ АВТОРАХ

Миронов Валерий Викторович, проф. каф. автоматиз. систем упр-я. Дипл. радиофизик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. сист. (УГАТУ, 1995). Иссл. в обл. иерархич. моделей и ситуац. управления.

Маликова Карина Эмильевна, асп., асс. той же каф. Дипл. информатик-экономист (УГАТУ, 2007). Готовит дис. в обл. использования встроенных моделей в интернет-приложениях.