

ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ ТРЕХМЕРНЫХ И ДВУМЕРНЫХ WEB-ТЕХНОЛОГИЙ

Зори С.А., Ковальский С.В.

Кафедра Прикладной математики и информатики,
Донецкий Национальный технический университет
zori@pmi.dgtu.donetsk.ua, skovalsky@yandex.ru

Abstract

Zori S.A., Kovalsky S.V. Organization of 3-dimensional and 2-dimensional web-technologies interaction. An approach to receiving text information by VRML-application is given. Procedures of interaction between HTML-document and VRML-application and procedures of retrieving data by VRML-application from CGI-application are described.

Введение

Интенсивное развитие средств аппаратного ускорения процессов вычислений и обработки графических данных ведет к расширению сферы применения трехмерной графики и все более интенсивному ее использованию в Internet-приложениях. Наиболее популярным языком описания трехмерной графики для Web является Virtual Reality Modeling Language (VRML), а общепринятым ISO-стандартом этого языка – версия VRML97. Основными преимуществами VRML являются простота и гибкость описания виртуальной модели мира.

Однако гибкость языка имеет обратную сторону: стандарт VRML97 не предполагает наличия ни определенных устройств ввода (таких как “мышь” или клавиатура), ни двумерного устройства вывода (монитора, принтера и т.п.) [1]. Наиболее распространенные средства просмотра VRML – встраиваемые в Web-браузеры модули – предлагают использовать “мышь” для навигации и выполнения операций над объектами мира. Поскольку стандарт языка VRML не предполагает наличия двумерного устройства вывода, то организация двумерного интерфейса в плоскости экрана становится весьма сложной задачей, решение которой в значительной степени зависит от аппаратного и программного окружения, обеспечивающего навигацию по миру.

Одной из основных проблем при построении двумерного интерфейса становится организация ввода текстовой информации, поскольку язык VRML не предполагает наличия клавиатуры как средства ввода. Наиболее очевидное решение данной проблемы – создание виртуальной клавиатуры, нажатие клавиш которой осуществляется “мышью” – не может быть признано удовлетворительным по критериям удобства и скорости ввода для пользователя, а также является весьма объемным для программирования. Дополнительные трудности возникают в связи с отсутствием у большинства средств просмотра VRML-миров многоязыковой поддержки.

В данной статье рассматриваются вопросы организации обмена данными между VRML-приложением и HTML-документом, передачи текстовой информации в виртуальный мир, а также построение клиент-серверных VRML-приложений с использованием CGI-интерфейса.

1. Применение языка HTML как средства ввода текстовой информации для виртуальных миров

Средствами собственно языка VRML невозможно создать удобный и надежный механизм ввода текстовой информации, поэтому необходимо использовать другие Web-технологии, которые могли бы обеспечить достаточно простую процедуру ввода данных пользователем и передачи введенной информации в виртуальный мир.

Автором предлагается использовать для организации двумерного интерфейса язык HTML, а для передачи данных из двумерного интерфейса в трехмерный – средства скриптовых языков JavaScript и ECMAScript, встроенных в HTML- и VRML-документы соответственно. Предлагаемая структура Web-приложения, объединяющего документы HTML и VRML приведена на рис. 1. Документ HTML может включать в себя виртуальный мир созданный средствами VRML как внедренный объект или как один из документов фреймовой структуры.

На рис. 2 приведен исходный код функции на языке ECMAScript, обеспечивающей выполнение набора инструкций JavaScript в охватывающем виртуальный мир документе HTML. Узел Inline языка VRML предназначен для включения в главный VRML-файл кода из других VRML-файлов, расположение которого указывает поле url. Если значением поля url узла Inline будет строка вида "javascript:...", то результатом создания узла станет выполнение инструкций языка JavaScript. Используемый в функции JSExec метод createVrmlFromString объекта Browser позволяет динамически формировать узел Inline, выполняя при этом необходимые инструкции JavaScript в нужный момент времени. Таким образом, выполняется передача данных и команд от виртуального мира двумерному интерфейсу. Однако результаты выполнения инструкция JavaScript недоступны для виртуального мира, поэтому для получения введенной пользователем информации необходимо использовать в качестве посредника серверное приложение, использующее CGI-интерфейс. Функция CGI-сценария при этом заключается в получении от двумерного интерфейса введенных данных и хранении до тех пор, пока от виртуального мира не придет запрос на их получение.

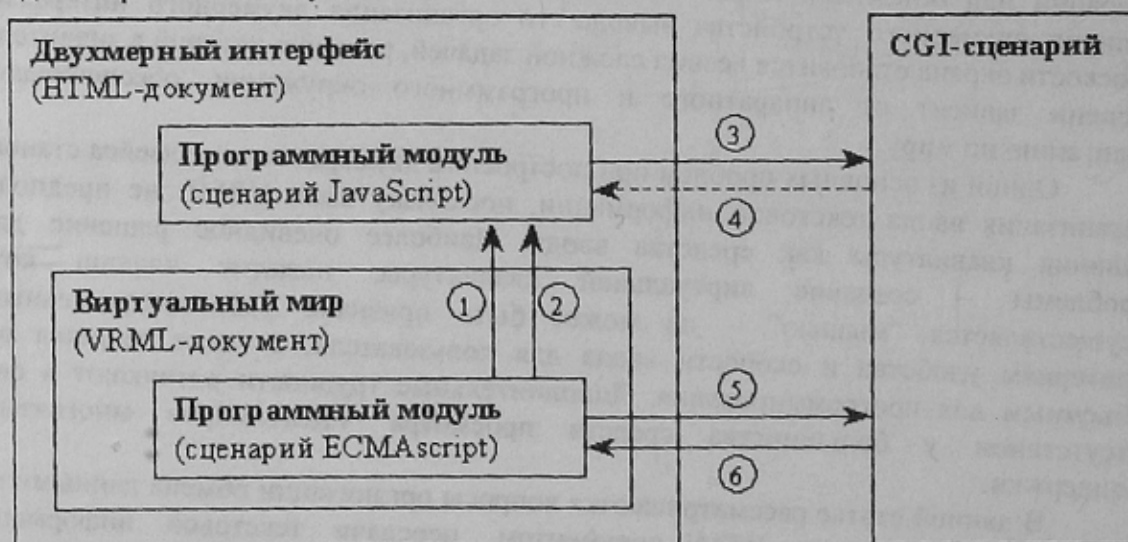


Рисунок 1 – Схема взаимодействия двумерного интерфейса и виртуального мира

Взаимодействие VRML- и HTML-документов предполагает сеансовую работу механизма ввода. В течение одного сеанса каждый из сценариев выполняет определенные действия (рис. 1).

Действия сценария ECMAScript:

- 1) инициализация сеанса ввода – передача запроса на ввод необходимой информации в двумерный интерфейс (1);
- 2) переход в режим периодического опроса CGI-сценария (5) и (6);
- 3) получение от CGI-сценария результатов ввода (6);
- 4) завершение сеанса ввода и уведомление, при необходимости, двумерного интерфейса о завершении сеанса (2).

Действия сценария JavaScript:

- 1) получение запроса на ввод необходимой информации (1);
- 2) принятие вводимых пользователем данных;
- 3) передача введенных данных в CGI-сценарий (3), ответ на CGI-запрос (4) при этом не несет смысловой нагрузки и игнорируется сценарием JavaScript.

Действия CGI-сценария:

- 1) формирование ответов (6), содержащих подтверждение продолжения ожидания, на периодические запросы сценария ECMAScript (5);
- 2) получение и сохранение данных, переданных сценарием JavaScript (3), поскольку ответ (4) не несет смысловой нагрузки, то он должен быть максимально коротким;
- 3) передача в ответ на очередной запрос сценария ECMAScript (5) введенной пользователем информации (6).

```
function JSExec(JSString)
{
  newNode =
  Browser.createVrmlFromString(
    'Inline { url ["javascript:'+JSString+""] }'
  );
  newNode = null;
}
```

Рисунок 2 – Исходный код функции JSExec

```
ScriptURL = new MFString(
  "/cgi-bin/main.pl?login="+UserLogin+
  "&pass="+UserPass+
  "&sender=Response"+
  "&ResponseNumber="+ (ResponseNumber+1) );
Browser.createVrmlFromURL (ScriptURL, MainScript,
  "serverScript");
```

Рисунок 3 – Выполнение CGI-запроса на языке ECMAScript

Выполнение CGI-запроса виртуальным миром (рис. 3) осуществляется при помощи метода `createVrmlFromURL` объекта `Browser` – этот метод выполняет компиляцию набора узлов из кода VRML, находящегося по заданному адресу. Поскольку в качестве адреса можно использовать строку CGI-запроса, то при помощи данного метода реализуется получение динамически создаваемого VRML-кода, несущего в себе ответ на запрос.

2. Использование CGI-интерфейса при построении VRML-приложения

В предложенном способе взаимодействия HTML-документа и VRML-приложения CGI-интерфейс обеспечивает один из этапов передачи данных. В отличие от обычного способа применения, когда ответ CGI-приложения на запрос полностью заменяет исходный HTML-документ, в данном случае предлагается использовать CGI-запрос для получения от серверного приложения фрагмента программного кода, который бы временно встраивался в локальное VRML-приложение, передавал необходимые данные и после завершения передачи исключался бы из состава приложения. Порядок действий по формированию и работе программного кода, передающего данные в VRML-приложение, приведен на рис. 4.

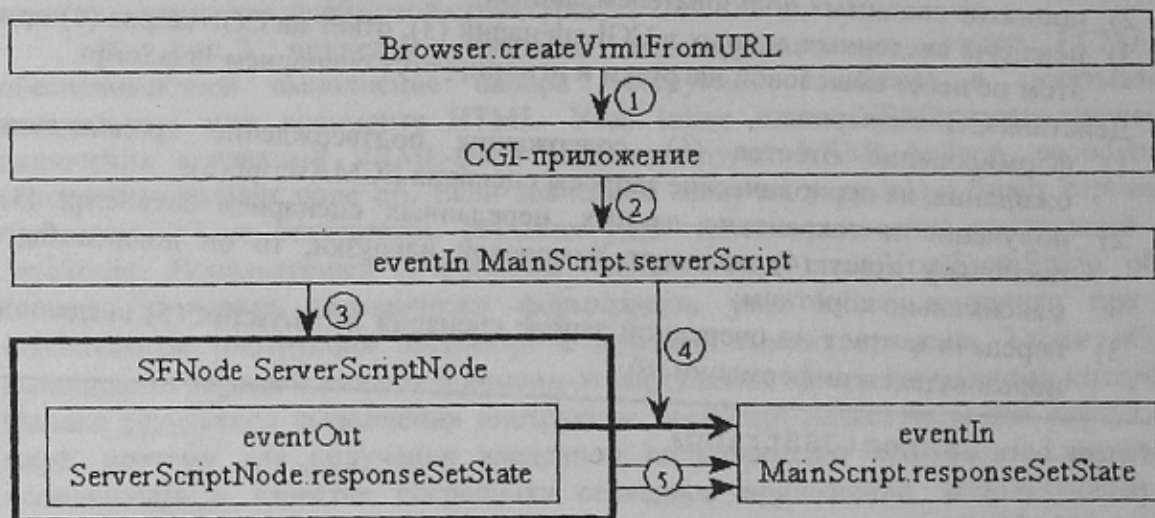


Рисунок 4 – Схема загрузки данных VRML-приложением с сервера

Обращение к серверу инициируется вызовом метода createVrmlFromURL объекта Browser (1). CGI-приложение передает в ответ на запрос фрагмент кода VRML, формируемый на основе содержимого базы данных. На рис.5 приведен шаблон ответа на CGI-запрос.

Возвращаемый CGI-приложением код содержит описание узла Script, который имеет одно обязательное исходящее событие responseSetState и некоторый набор дополнительных исходящих событий. ECMAScript-код этого узла содержит описание функции initialize, содержащей инструкции, обеспечивающие посылку всех исходящих событий узла.

Метод createVrmlFromURL передает сформированный на основе полученного кода узел входящему событию MainScript.serverScript (2). Обработчик данного события присваивает полученный узел глобальной объектной переменной ServerScriptNode (3) и устанавливает маршрут (4) между одноименными исходящим и входящим событиями responseSetState узлов ServerScriptNode и MainScript. Сразу после окончания работы обработчика события serverScript начинается выполняться функция initialize узла ServerScriptNode которая генерирует исходящее событие, проходящее по созданному маршруту. Обработчик входящего события MainScript.responseSetState в зависимости от полученного значения выполняет различные действия, среди которых может быть чтение значений других исходящих событий узла ServerScriptNode (5).


```

Content-type: x-world/x-vrml

#VRML V2.0 utf8
script {
  eventOut SFInt32 responseSetState
  // список других исходящих событий
  .
  uri "javascript:
      function initialize()
      {
        responseSetState=1;
        // установка значений других исходящих событий
        .
      }
  "
}

```

Рисунок 5 – Шаблон возвращаемого CGI-приложением VRML-кода
Предлагаемый способ выполнения CGI-запроса VRML-приложением является универсальным и может использоваться для организации взаимодействия виртуального мира с любыми другими Web-технологиями, использующими CGI-интерфейс.

3. Результаты практического применения разработанных методов

Описанные методы были использованы при разработке автором системы моделирования наружной рекламной кампании. Разработанная система предназначена для моделирования физического размещения средств наружной рекламы в городе. Система является многопользовательской, данные пользователей хранятся на сервере. Работа системы на машине-клиенте обеспечивается двумерным (рис. 6) и трехмерным (рис. 7) интерфейсами. Описанные методы передачи текстовой информации применялись для передачи имен проектов, путей к файлам изображений и т. п.

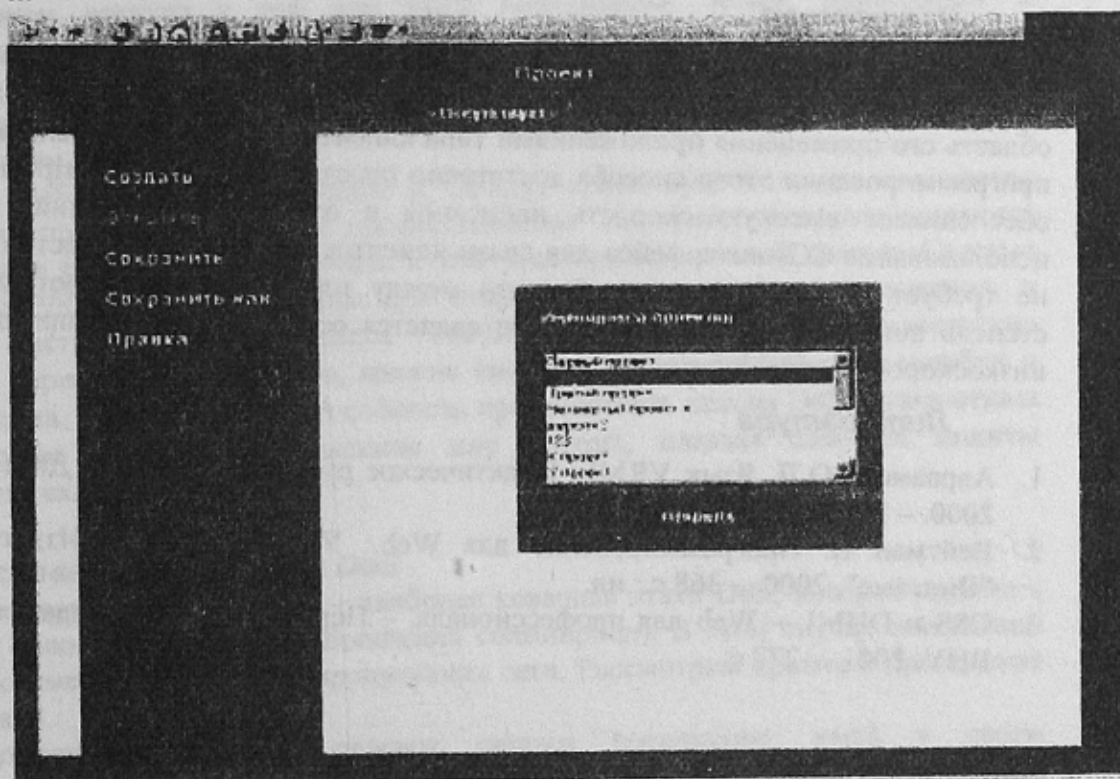


Рисунок 6 – Экранная форма двумерного интерфейса

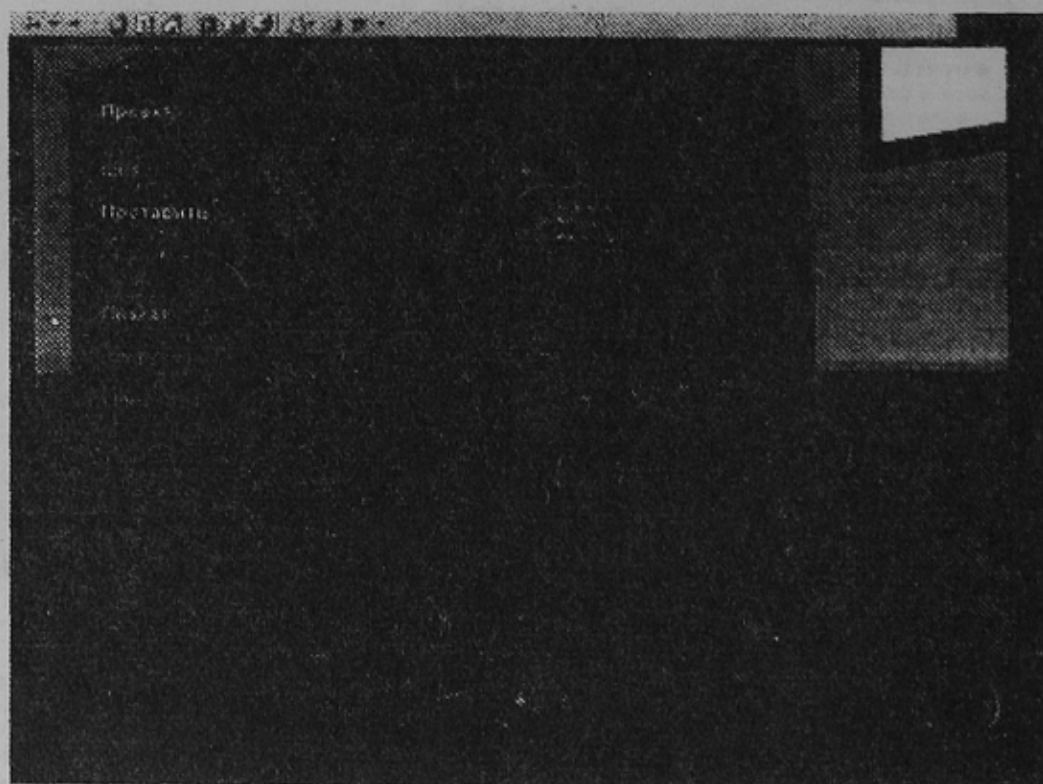


Рисунок 7 – Экранная форма трехмерного интерфейса

Доступ к системе открыт по адресу www.udkr.donbass.com/VirtualCity/.
Регистрация пользователя и работа с системой являются бесплатными.

Заключение

Использование CGI-интерфейса является необходимой частью приведенного способа организации двумерного интерфейса виртуального мира, что ограничивает область его применения приложениями типа клиент-сервер. Однако использование при программировании этого способа достаточно простых языков JavaScript и ECMAScript обеспечивает высокую скорость написания и отладки приложений. Кроме того, использование CGI-интерфейса для связи клиентской и серверной частей приложения не требует наличия постоянной связи между ними и обеспечивает значительную степень автономности их работы, что является особенно важным при эксплуатации низкоскоростных каналов связи.

Литература

1. Авраамова О.Д. Язык VRML. Практическое руководство. – М.: ДИАЛОГ-МИФИ, 2000. – 288 с.
2. Вейтман В. Программирование для Web.: Уч. пос.: – М.: Издательский дом "Вильямс", 2000. – 368 с.: ил.
3. CSS и DHML – Web для профессионала. – Пер. с англ. – К.: Издательская группа BHV, 2001. – 272 с.