УДК 681.3.06

Б. С. Долговесов [1], Б. С. Мазурок [2], В. Г. Ванданов [3]

Институт автоматики и электрометрии СО РАН
пр. Академика Коптюга, 1, Новосибирск, 630090, Россия

Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия
E-mail: [1] bsd@iae.nsk.su; [2] boris@iae.nsk.su; [3] victor@iae.nsk.su

ИНТЕРАКТИВНАЯ СИСТЕМА ВИЗУАЛИЗАЦИИ ДЛЯ WEB ПРИЛОЖЕНИЙ

Для организации интерактивного взаимодействия между 3D сценой и Web приложением используется система запросов к глобальной базе данных. Сведение межмодульного обмена к обращению к глобальной базе данных позволяет унифицировать межмодульный обмен и ввести максимально полный контроль над системой в целом. Основа для унификации – специально разработанный язык запросов – XQL скрипт (eXtended Query Language).

Ключевые слова: Web3D, XQL, 3D visualization systems.

Введение

В связи с тем, что структура 3D сцен и протоколов обмена непрерывно усложняется, возникла необходимость в новом подходе к организации хранения и передачи данных. Основная идея заключается в том, чтобы ввести единый механизм как для модификации базы данных, так и для передачи данных по сети и в целом для всего межмодульного обмена.

В таком подходе любой модуль системы через запросы к базе данных, получает доступ к данным от любого источника независимо от его типа – данные могут поступать от удаленного компьютера, управляющего комплекса, манипулятора, скрипт-интерпретатора или просто из файла. Таким образом, любой модуль получает универсальный доступ практически к любым параметрам системы – таким, как положение объектов, состояние анимации, параметры растрирования – вплоть до цвета отдельной грани, или shader параметров GPU.

Особенно актуален такой подход при внедрении 3D сцен на HTML страницы Web приложений, поскольку позволяет простыми средствами полностью контролировать создание и модификацию 3D объектов.

Область применения системы включается в себя такие Web приложения, как энциклопедии, справочники, Интернет университеты, магазины, биржи и т. п. Интерактивные 3D модели могут использоваться для учебных демонстраций, демонстраций товаров, визуализации бизнес-процессов (3D графики и гистограммы), визуализации данных научных исследований и пр.

Принципы построения системы

Основные требования к системе можно сформулировать в следующем виде:

- совместимость – добавление новых структур данных не должно приводить к несовместимости со старыми структурами;
- гибкость – возможность задания произвольных структур данных;
- простота – минимальное количество специальных требований;
- универсальность – унифицированные способы обмена данными для разных типов источников: сеть, файл, мышь, джойстик и т. п.

Традиционный подход к управлению сценой заключается в предоставлении интерфейса к объектам сцены – набора команд, которые позволяют контролировать поведение объектов. Как правило, этот набор сильно ограничен и позволяет менять только самые простые свойства объектов – положение в сцене и проигрывание анимации. Таким образом организованы, например, системы iSpace (Caligari Corporation); Conductor i3Di (Cybelius Software); Macro-

media Director 8.5 Shockwave Studio; Bryce, Poser, HyperView (Viewpoint) и др. [1] Кроме того, существующие форматы 3D сцен, либо хранят данные в бинарном виде, либо используют тэговый XML формат. Что в любом случае требует специального механизма для модификации данных – отличного от механизма загрузки.

Замена прямых вызовов на запросы к базе данных и унификация доступа предоставляет возможность более детального управления сценой и не требует дублирования кода по созданию объектов и кода по их модификации. Поскольку файл полностью описывает сцену, то ясно, что загрузка сцены из файла всегда выполняется максимально детально. Но если в результате унификации обмена управляющий модуль (например, скрипт-интерпретатор) будет использовать те же средства, что и загрузчик сцены, то он будет иметь и те же возможности, что загрузчик сцены, т. е. максимально полный доступ.

Таким образом, в результате анализа существующих систем был выявлен существенный недостаток – формат представления 3D данных принципиально отличается от синтаксиса команд модификации данных, что не позволяет осуществлять полноценный контроль и затрудняет интеграцию со скриптовыми языками программирования. Унификация протоколов по загрузке и модификации, на наш взгляд, решает эту проблему.

Формат представления данных

Основными объектами 3D сцены являются геометрические примитивы, материалы, текстурные карты, камеры и источники света. Объекты объединены в модели. Набор моделей составляет сцену. Сцены отображаются в определенные области окна – порты, окна в свою очередь располагаются на одном или нескольких мониторах. Другими словами, в общем случае все объекты представляют собой сложный граф с многочисленными перекрестными связями.

Один из вариантов описания такого графа – введение набора связанных таблиц, где каждая таблица – это данные определенного типа: mesh, camera и т. п. Каждая строка таблицы представляет собой конкретный элемент определенного типа. Поиск по таблице может осуществляться стандартным способом аналогично SQL запросам по маске поиска, по имени объекта или по значению какого-либо поля.

Однако в таком подходе имеется следующие недостатки.

1. Поиск объекта по имени может дать неоднозначный результат, поскольку при формировании сцены из нескольких моделей, имена объектов могут совпасть. Например, при загрузке описания двух автомобилей в каждом из них может оказаться объект Wheel, Mirror, Camera и др.

2. Поскольку в таблице все поля равноправны, то поиск по любому полю происходит с одинаковой скоростью. Но реально объекты сцен отображаются в определенном порядке, например, сначала отображаются все модели и объекты одной сцены, потом объекты следующей сцены и т. д. Поэтому сканирование данных можно ускорить, если соответственно их сгруппировать.

В XQL базе данных, для оптимизации процесса поиска и однозначной идентификации, объекты располагаются в виде дерева в соответствии с иерархией принадлежности, например: SCENE-MODEL-NODE или WINDOW-PORT и т. п. В результате у каждого объекта появляется новый идентификатор – путь в дереве данных, который в отличие от имени уже однозначно определяет объект [Долговесов и др., 2006].

Выбранный синтаксис идентификатора пути похож на JavaScript как более привычный для разработчиков Web приложений:

```
SCENE.SceneA.MODEL.ModelB.NODE.NodeC
```

или `SCENE[SceneA].MODEL[ModelB].NODE[NodeC]`

Пример адресации объекта на удаленном компьютере:

```
HOST[10.0.0.5].SCENE.A.MODEL.B.IMAGE.C
```

[1] *Dolgy E.* 3D Internet: ждём больших перемен (http://www.inter.net.by/technologies/fridman02.html); *Фридман В.* Трехмерные миры в Web: VRML (http://www.fcenter.ru/online.shtml?articles/software/internet/494); Web3D Consortium (http://web3d.org).

Наполнение объекта данными производится простым присваиванием ему массива чисел и строк, в стиле JavaScript.

Например: `NODE.Car.POS=[100,0,0]` или `MESH.Car.MATERIAL = Red.`

Текстовое представление чисел дает возможность редактировать файл в любом текстовом редакторе и позволяет существенно уменьшить размер файла по сравнению со многими бинарными форматами по двум причинам:

- точность чисел регулируется простым заданием нужной длины;
- текстовое представление в среднем дает больший коэффициент сжатия при архивировании в ZIP файл.

Таким образом, XQL формат представления 3D данных, представляет собой архивированное текстовое представление 3D сцены.

Основные отличия XQL от существующих форматов представления данных:

1) формат XQL – компактное представление данных с необходимой точностью;

2) синтаксис XQL позволяет не только описывать данные, но и модифицировать их;

3) синтаксис XQL запросов совместим с синтаксисом наиболее распространенных скриптовых языков программирования JScript и VBScript.

В настоящее время существуют XQL экспортеры для наиболее распространенных редакторов 3D сцен, таких как 3DSMax и Maya [2]. Также система поддерживает импорт популярных форматов представления 3D данных VRML, DXF [3].

Организация обмена

Ключевым моментом в работе с базой данных являются плагины. При каждом обращении к базе данных выполняется последовательная обработка новых данных набором плагинов, зарегистрированных на обрабатываемый узел. Основное свойство плагина – поддержка двух функций: формирование данных при запросе на чтение и обработка данных при запросе на запись. Таким образом, запись в базу данных может породить множество вторичных обращений, которые могут производиться плагинами в процессе обработки запроса. Именно система распределения плагинов определяет поведение системы в целом.

Поскольку плагины имеют возможность взаимодействовать с другими аппаратно-программными устройствами, то именно через них и осуществляется обмен базы данных с внешним миром – 3D акселераторами, манипуляторами, джойстиками, мышью и пр. Такой обмен в конечном счете приводит к формированию изображения и пополнению базы даны новыми данными (структура обмена приведена ниже).

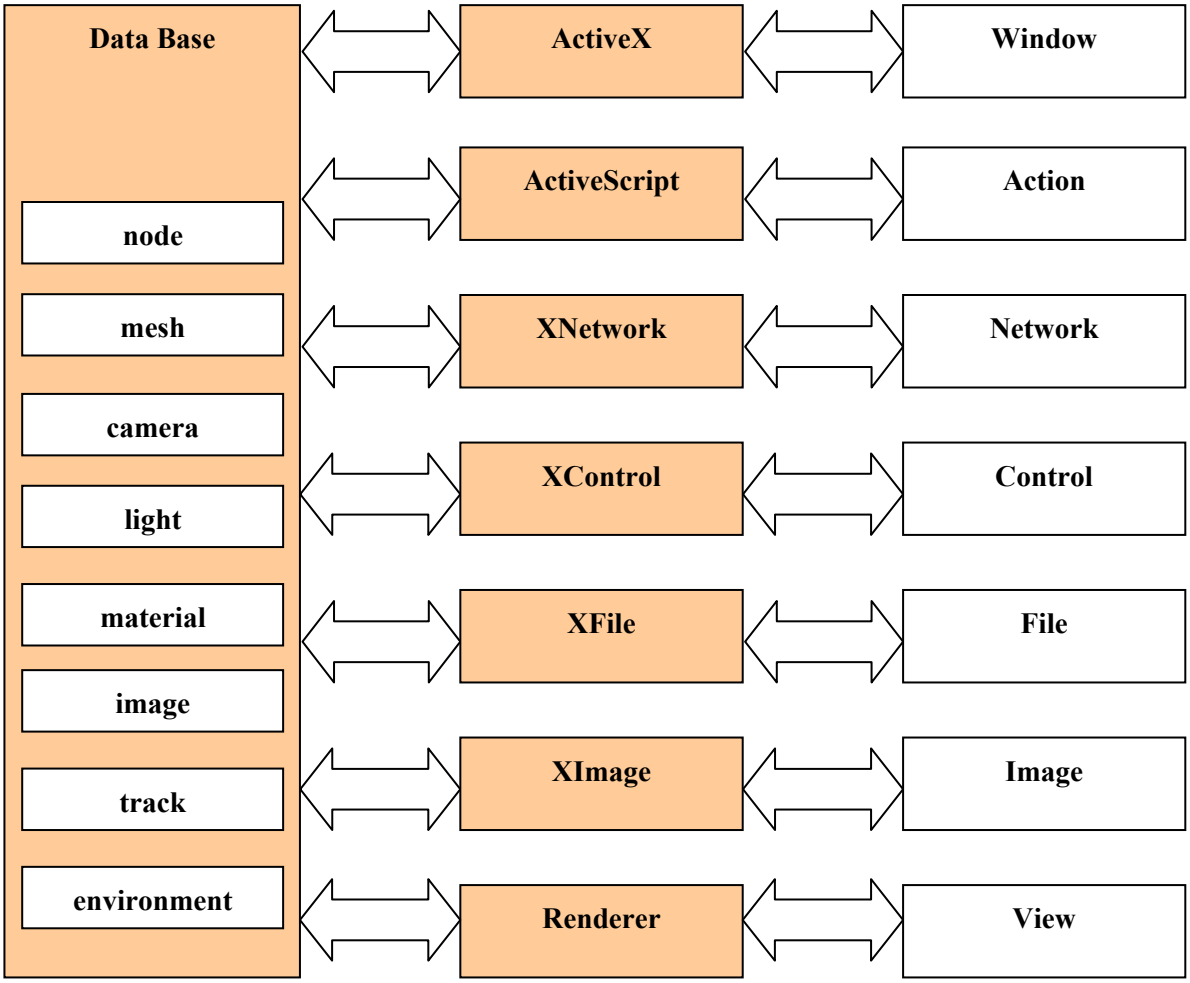
На схеме хорошо видно, что модули практически не взаимодействуют напрямую. Весь обмен идет через базу данных. Это позволяет распределить и распараллелить работу по созданию программного обеспечения. Единственное требование к этим модулям – все они должны поддерживать единый протокол обмена. Это позволяет отлаживать модули независимо и собирать их вместе для получения разных конфигураций системы.

Важным требованием для Web приложений является требование совместимости данных. Новые версии программы должны в полном объеме поддерживать отображение сцен и исполнение скрипт-программ, написанных пользователями ранее, и наоборот, в случае отказа клиента обновить программу, новые версии контента не должны разрушать работу системы.

Для обеспечения совместимости данных применяется достаточно мелкое их дробление на отдельные составляющие. В результате при возникновении новых структур данных те модули, которые еще не умеют работать с этими структурами, просто их игнорируют, но при этом не отказываются работать вообще. При этом неизмененные структуры данных обрабатываются так же, как и прежде.

[2] Autodesk 3DS Max (http://usa.autodesk.com); Autodesk Maya (http://usa.autodesk.com).

[3] Open Standards for Real-Time 3D Communication (http://www.web3d.org/x3d/vrml); Autodesk DFX Format (http://www.autodesk.com/techpubs/autocad/acad2000/dxf/dxf_format.htm).



Передача 3D данных по сети осуществляется специальным плагином, который синхронизирует базы данных на разных компьютерах и транслирует запросы.

Для внедрения в HTML среду разработан ActiveX компонент [4], который динамически подгружает необходимые модули и в случае необходимости автоматически обновляет компоненты с сервера (Live Update).

Разработанная система XQL запросов интегрирована с распространенными скриптовыми языками программирования JScript, VBScript и др. – на основе входящей в состав Windows, компоненты ActiveScript [5]. Интеграция выполнена на уровне внедрения в контекст ActiveScript машины COM объекта (с символьным именем «db»), предоставляющего доступ к глобальной базе данных системы.

Таким образом, все объекты базы данных являются либо объектами примитивного типа (INT, FLOAT, BSTR и т. п.), что соответствует листьям дерева базы данных, либо указателями на другие, дочерние по отношению к «db», COM объекты – узлы дерева базы данных.

При изменении командами скрипта свойств «db» выполняется запись в базу данных системы и производится стандартная отработка изменений соответствующими плагинами. При чтении свойств «db» возвращаются данные из базы данных системы.

Например: `var fov = db.CAMERA.Cam.FOV`

Такой подход позволяет не только использовать стандартные языки программирования для моделирования внешнего вида виртуальных объектов, но и программировать сколь угод-

[4] *Cluts N.* Microsoft ActiveX Controls Overview (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaxctrl/html/msdn_actxcont.asp).

[5] Microsoft Windows Script Technologies (http://msdn2.microsoft.com/en-us/library/d1et7k7c(VS.85).aspx).

но сложное их поведение. Также система предоставляет механизмы для динамического создания 3D объектов, процедурных 3D объектов – такие, как 3D графики, гистограммы и т. п.

Для разработки графического интерфейса Web приложения и отладки скрипт-программ можно использовать стандартные интегрированные средства разработки, такие как Microsoft® Visual Web Developer 2005 Express Edition, Adobe® Dream Waver и др.

Заключение

В работе проведен анализ требований к интерактивной системе визуализации для Web приложений и предложен вариант реализации 3D системы визуализации, основанный на унифицированных запросах глобальной базы данных с помощью стандартных скрипт-интерпретаторов JScript и VBScript.

Список литературы

Долговесов Б. С., Мазурок Б. С., Ванданов В. Г. Разработка новых и адаптация существующих базовых алгоритмов визуализации 3D сцен для Web приложений // Труды 16-й Междунар. конф. по компьютерной графике и ее приложениям «Графикон-2006» (Новосибирск, Россия, 1–5 июля 2006). Новосибирск, 2006. С. 317–319.

Материал поступил в редколлегию 12.05.2008

B. S. Dolgovesov, B. S. Mazurok, V. G. Vandanov

Interactive Visualization System for Web Applications

To organize the interaction between interactive 3D scenes and Web applications system queries to the global database used. Reducing intermodule communication to query to the global data base allows one to unify intermodule exchange and provides full system control. The basis for unification is specially developed query language - XQL script (eXtended Query Language) [1].

Keywords: Web3D, XQL, 3D visualization systems.