

# Языки Описания Видов Отображения для Систем Компьютерной Визуализации

В.Л. Авербух, Н.И. Бараковских, А.И. Зенков А.Н. Петров  
averbukh @ imm.uran.ru  
ИММ УрО РАН, УрГУ

Работа выполнена при поддержке гранта РФФИ № 04-07-90120.

В работе описывается подход к реализации «систем, генерирующих специализированные системы визуализации», а также примеры их реализации и языки, являющиеся их основой.

## 1. Введение

Следует отметить две тенденции развития систем визуализации. С одной стороны – разработка универсальных средств визуализации, а с другой – специализацию по всем направлениям, вплоть до создания специальных графических станций с реализацией для данного случая графическим алгоритмическим и программным обеспечением.

Разработка специализированных систем визуализации необходима в тех случаях, когда пользователя интересуют те особенности, интересные явления и процессы, изучение которых не обеспечивается за счет стандартных методов отображения. Тогда требуется создание новых способов графического описания явлений и процессов или их моделей. (Иногда эти способы являются комбинацией уже известных или базируются на естественной или привычной образности, характерной для изучаемого явления.)

Можно говорить о видах отображения, стандартных или придуманных для данного случая методиках визуального представления данных, своего рода, визуальных процедурах, которые при реализации в конкретных визуальных средах и при подстановке реальных данных выводятся на те или иные графические устройства. Кроме этого, в такой «процедуре» (то есть в виде отображения) могут предусматриваться возможные изменения изображений, включая анимацию, и допустимые способы взаимодействия с картинкой.

Специализированные системы визуализации характеризуются своим набором специфических видов отображения, используемых в тех случаях, когда стандартные методы не дают результата.

Именно создание новых видов отображения является в настоящее время наиболее непредсказуемой частью процесса разработки специализированных систем, тогда как программы, непосредственно визуализирующие необходимые объекты зачастую сходны по своей структуре. (Хотя сами по себе содержат очень сложные в алгоритмическом плане и плане реализации компоненты.)

Появилась идея разработки нового типа систем, занимающихся генерацией специализированных систем в рамках некоторого класса за счет описания видов отображения. Таким образом, системы порождают набор конкретных (часто анимационных) графических выводов, обеспечивающих требования пользователей.

Основной специализированных систем визуализации можно считать набор видов отображения, призванных отображать необходимые для данного случая особенности и интересующие пользователя процессы в тех случаях, когда стандартные методики визуализации не дают результата. Дав пользователю язык описания видов отображения и реализовав на его базе генерацию

конкретных систем, мы ускоряем и облегчаем разработку программных продуктов.

Ниже рассмотрены три примера подобных «систем, генерирующих системы».

## 2. Среда разработки специализированных систем научной визуализации

В сотрудничестве с математиками был разработан набор систем, хорошо учитывающих специфику задач и ориентированных на выделение важных для исследователя особенностей исследуемых математических объектов [1-3].

Существенным результатом этого сотрудничества можно считать созданный единый вычислительно-визуализационный комплекс, который позволил получить новые результаты в предметной области математики.

Однако отсутствие единого подхода к разработке программного обеспечения приводило к тому, что при реализации каждой новой постановки задачи приходилось заново создавать визуализационный модуль и реализовывать его связь с вычислительным модулем. Эта проблема и послужила толчком к попытке унифицировать разработку специализированных систем визуализации.

Унифицированная система должна содержать модуль визуализации, общий для различных специализированных систем, и набор модулей, обеспечивающих описание и восстановление трехмерных сцен с учетом особенностей конкретных математических объектов. Для визуализации и исследования новых объектов пользователь-математик может разрабатывать собственные модули. Форматы входного потока данных такого модуля должны быть согласованы с соответствующей счетной программой, а выходного - с модулем визуализации.

Анализ требований пользователей, предъявленных ими при проектировании разработанных систем, показывает, что для целей построения унифицированной системы весьма удобен язык, основанный на фотореалистической трехмерной графике. Такой язык определяется набором возможных видов отображений с изменяемыми атрибутами отдельных объектов. Структурными единицами его являются геометрические конструкции из трехмерных тел и поверхностей, отображаемых на экране с учетом их атрибутов (цвет, освещенность, прозрачность и т.д.). Основная задача при создании проекта системы заключается в том чтобы, обеспечить максимальную гибкость и настраиваемость системы, избежав при этом функциональной избыточности и сложности использования. Для достижения этой цели была использована идея построения системы в виде набора модулей, взаимодействующих между собой по заранее определенной схеме и оговоренным форматам данных. Архитектура системы состоит из 2х основных частей:

1. Подсистема обработки исходных данных
2. Модуль визуализации.

## 2.1 Подсистема обработки исходных данных

Это наиболее гибко настраиваемая часть системы, в которой сосредотачивается основная работа по созданию конкретной ее реализации. Назначение данной подсистемы – из файлов с исходными данными получить описание 3-х мерной сцены.

При этом необходимо:

- Обеспечить простое использование типовых алгоритмов 3-х мерной компьютерной графики (триангуляция поверхностей, децимация поверхностей, построение нормалей, и т.д.)
- Дать пользователям декомпозировать их собственный процесс создания сцены.

Это достигается за счет введения понятия модуля. Модуль – это часть общего процесса построения сцены, которая выделяется в отдельный алгоритм и может использоваться независимо.

Для всех конфигурационных файлов используется формат XML. Для файлов другого типа может быть описана своя цепочка. Таким образом, при запуске системы на конкретном файле с данными производится поиск соответствующей цепочки в конфигурационном файле и если она найдена, последовательно вызываются все модули из цепочки. Результат работы предыдущего передается на вход следующему. При этом в промежуточных файлах контролируется только их тип, чтобы гарантировать, что следующий модуль сможет работать с этим типом. Регламентируется только формат итогового файла. Это должно быть описание трехмерной сцены.

Следует отметить, что модули - это и есть существенная часть процесса визуализации в разрабатываемой системе. Алгоритмы модулей определяют метафору и способы визуализации конкретных математических объектов. Поэтому модули должны создаваться при непосредственном участии математиков-пользователей системы.

Окончательно сформированная сцена перед поступлением в визуализатор записывается в файл в определенном формате. На данный момент реализовано 2 формата. Первый – текстовый формат на основе XML, который используется пока для тестовых целей, а в перспективе предназначен для реализации связи с форматом VRML. Второй – двоичный формат. Хорошо подходит для хранения больших сцен, так как данные хранятся очень компактно. Имеет такую же структуру, как и первый формат, только вместо тэгов использует специальные открывающие и закрывающие сигнатуры. Оба формата очень просты и удобны для пользователей.

## 2.2 Визуализатор

Непосредственно вывод изображения и работу с изображением в интерактивном режиме осуществляет программа «визуализатор».

Этот модуль является наиболее трудоемким с точки зрения программирования, так как он должен обладать достаточно развитой функциональностью и обеспечивать интерактивную работу пользователя. Ранее при построении систем визуализации “с нуля” вся работа с исходными данными и непосредственно вывод изображения реализовывались каждый раз заново. Основной проблемой являлось создание мощной и удобной программы интерактивной работы с изображением. При изложенном подходе при разработке новой системы уже имеется готовый

модуль, который независимо может совершенствоваться и расширяться.

Модуль визуализации реализует пользовательский интерфейс управления трехмерными объектами и процедуру считывания наборов трехмерных объектов из файлов-описаний сцен, созданных ранее. Он обеспечивает интерактивные средства управления выводом изображения и изменения видов отображения информации. Пользователь имеет возможность быстро изменять положение и ориентацию объекта в трехмерном пространстве, масштаб вывода изображения, количество и положение источников света и секущих плоскостей, ракурс вывода, а так же отображать объекты в различных режимах прозрачности.

На данный момент система находится в стадии опытной эксплуатации. На базе описываемого подхода реализованы системы визуализации для нескольких задач, связанных с решением проблем теории игр и оптимального управления. Большие результаты получены при построении визуализации множества функции цены и сингулярных поверхностей для задачи теории дифференциальных игр. Принципы специализированности, заложенные в предлагаемом подходе, позволили получить и увидеть такие детали поверхностей, которые невозможно было увидеть при помощи универсальных систем визуализации. Эти детали дали возможность существенно продвинуться в исследовании данной проблемы. Также сделан первый вариант визуализации для задачи оценивания множеств достижимости линейных многошаговых систем с фазовыми ограничениями. Специфика этой задачи заключается в многомерности получаемых геометрических объектов. Сейчас вырабатываются дополнительные виды отображения для эффективной работы с такими объектами.

Также на данный момент несколько новых возможностей системы находятся в завершающей фазе: редактируемая история, сохранение сцены, возможность создания параллельных иерархий объектов с помощью иерархических тэгов.

Основой модуля отображения трехмерной графики является высокоуровневая библиотека Java3D фирмы Sun. Выбор этой библиотеки позволил значительно уменьшить время, затраченное на разработку и получить возможность использовать современное оборудование для виртуальной реальности. Java3D также является многоплатформенной библиотекой, что значительно расширяет круг пользователей системы и делает возможным ее использование через WWW.

## 3. Визуализация биологических моделей

В ходе разработки ряда систем визуализации отображающих модели биологических явлений (в частности, кинетики размножения и мутирования вирусов и рециркуляции кальция в кардиомиоците) было замечено, что множество операций повторяются от реализации к реализации. При этом они не связаны со спецификой задачи, а в основе своей являются рутинными операциями графического вывода (например, анимация объектов-примитивов, генерация событий в результате действий пользователя, связанных с тем или иным объектом, вычисление пресечения объектов), кочующими из проекта в проект. Отметим, что модели данной прикладной области сильно зависят от уровня описания конкретного биологического процесса. Часто со временем проявляется новый уровень детализации или же появляются новые, неизвестные ранее детали процессов. Программист зачастую имеет дело с

качественным описанием процесса, которое непросто перевести в анимационную последовательность.

Таким образом, появилась задача по созданию системы, которая бы позволяла программировать анимацию на уровне объектов конкретной предметной области, а не на уровне абстракций конкретного языка программирования. Система лишь определяет основные правила, по которым она «умеет играть», и если программист хочет добавить объекты специфичные для данной предметной области или задачи, то ему нужно создать объекты, удовлетворяющие требованиям системы.

Разработан некоторый «каркас», который предоставляет пользователю удобную и понятную архитектуру, основанную на компонентном подходе, дает возможность конечному пользователю создавать и добавлять новые компоненты. Также создан метаязык, который позволяет иметь дело лишь с текстовым описанием элементов, то есть отказаться от прямого программирования с использованием классов каркаса.

При этом решено сделать каркас для двумерной анимации, так как в настоящее время рассматриваются модели принципиально требующие именно двумерности. В дальнейшем предполагается перейти к построению каркаса для трехмерной анимации

Метаязык – это вспомогательное средство, необходимое для еще большего упрощения использования каркаса. Метаязык описания анимации включает в себя описание объекта, его поведения, событий, которые с ним могут произойти и т.п.

В качестве инструмента реализации был выбран язык XML, который показался наиболее удобным для построения нашего языка, с точки зрения удобства, широкой поддержки практически во всех языках программирования, в частности в Java, которая была использована для разработки этого каркаса.

Следующей задачей является разработка методов генерации этого языка на основе качественного описания биологических процессов. Очевидно, что использование для этой цели естественного языка (в данном случае языка биологии и физиологии клетки и пр.) затруднительно. Поэтому предполагается использование шаблонов, с помощью которых пользователь сможет описать как сам сложный объект, так и его поведение, чтобы в результате появилась основа («сценарий») для создания анимационного фильма.

#### **4. Визуализация производственных процессов**

Обеспечение управления производства порождает отдельный класс вычислений. Как следствие, для этого класса вычислений требуется особый тип визуализации. Вполне очевидным является, то что, для обеспечения этого типа визуализации естественна анимация и трёхмерность. На данный момент развитие компьютерной графики позволяет использовать технологии связанные с виртуальной реальностью. Однако в условиях производства не имеет смысла использовать дорогостоящие средства «погружения» и пр., а следует обеспечить достаточный для данной задачи уровень реализма.

После реализации ряда специализированных систем визуализации, служащих для отображения производственных процессов в металлургических цехах была поставлена задача разработки системы, позволяющей достаточно просто и быстро создавать новые специализированные системы

Простая анимация, не зависящая от пользователя и от каких либо внешних воздействий, не подходит для визуализации производственных процессов.

Управление событиями в анимационном фильме должно идти за счет реальных данных получаемых по ходу работы тех или иных механических устройств. Для нормальной визуализации работы того или иного механизма также необходима возможность отражения реакции на различные действия оператора, сидящего за пультом управления. Возможно, что действиями механизма вообще управляет внешняя программа, для демонстрации или испытания её необходим интерактивный трёхмерный фильм, реагирующий, как на действия пользователя, так и на изменения во внешней среде.

Система должна быть расширяемой, то есть иметь возможность постоянного добавления тех или иных возможностей (в первую очередь это касается событийности модели). Требуется также возможность загрузки в «виртуальную вселенную» трёхмерных объектов некоторых стандартных форматов (3D MAX, Maya, VRML и т.д.) То есть система должна загружать набор трёхмерных объектов, а также должна обрабатывать какие-либо события, генерируемые кем или чем угодно, поставив им в соответствие некоторые поведения (набор действий по преобразованию объекта).

Основой среды разработки специализированных систем в случае визуализации производственных процессов стал предложенный и реализованный язык описания анимации, который позволяет быстро и достаточно просто описать как сам процесс, так и его временные характеристики.

Трёхмерная сцена описывается в виде дерева. Для описания древовидных структур естественно использовать XML. При этом описывается сам объект его поведение, освещенность, перемещение камеры и пр.

Основная функция, для описания анимации заставляет двигаться объект по заданной траектории (пути – набору кривых, заданных в параметрическом виде).

Естественно, объекты могут появляться по ходу работы системы (например, на склад пришёл новый контейнер). Для создания объекта динамически предусмотрена специальная функция.

Также как и объект динамически создаются новые пути, с добавлением к ним отдельных кусков. Также создан ряд функций для отладки написанных скриптов.

Помимо управляющего скрипта необходимо наличие описания множества объектов сцены и других вспомогательных вещей (таких как освящение, точки взгляда, пути). Всё это описывается в xml файле.

Трёхмерные объекты, хранящиеся в формате 3ds, отображаются на сцене с помощью функций библиотеки World Tool Kit (очень мощная библиотека по работе с трёхмерными изображениями, построенная на основе OpenGL).

Влияние пользователя на картинку по смыслу задачи должно быть минимальным. По этому, дана единственная возможность передвижения камеры по сцене с помощью мыши.

Каждую функцию скрипта система переводит в последовательность некоторых элементарных команд, которые заносятся в очередь. По ходу выполнения скрипт преобразуется в некий исполняемый код, который потом можно вызывать многократно, например, для записи демонстрационного ролика, правда в этом случае уже не будет никакого взаимодействия с внешней средой.

Возможности связи с внешней средой ограничены возможностями языка VB script, то есть вся связь происходит за счёт различных COM объектов.

В качестве инструментария была выбрана мощная библиотека по работе с 3D объектами – World Tool Kit от Sense8, одного из крупнейших разработчиков в области виртуальной реальности. (Проект изначально задумывался для внутреннего пользования, то есть для испытания и демонстрации программного обеспечения для различного оборудования, и проблемы легального использования и высоких системных требований не стояли). Данная библиотека имеет все возможности OpenGL, а также намного более удобные методы работы с трёхмерными объектами, и методы для работы с различными устройствами ввода и вывода, такими как мышь, трекбол, джойстик, экран, виртуальные шлемы перчатки, таким образом, при желании с помощью библиотеки можно создать более серьёзную виртуальную реальность. Хотя задача по поддержанию полноценной виртуальной реальности пока не стоит, но эксперименты по работе с более мощным инструментарием были вполне успешно проведены.

Дальнейшая работа может продолжаться в двух направлениях

- 1) Разработка пользовательского интерфейса для создания XML документа и скрипта.
- 2) На базе созданного языка создать ряд крупных моделей, в частности,

проекта, названного «виртуальный завод».

## 5. Заключение

Конвейер визуализации, который задает рамки проектирования систем визуализации, описывая этап за этапом разработки, необходимо дополняется важным этапом – этапом разработки системы видов отображения. Разработанные нами средства помогают если не полностью автоматизировать, то в значительной мере облегчить его и привлечь к нему пользователя-специалиста в конкретной области.

## Литература

1. Авербух В.Л., Зенков А.И., Исмагилов Т.Р., Манаков Д.В., Пыхтеев О.А., Юртаев Д.А. Разработка специализированных систем научной визуализации // Алгоритмы и програм. средства парал. вычислений: Сб. науч. тр. / ИММ УрО РАН.- Екатеринбург, 2000.- Вып.4. С.3-23.
2. Averbukh V.L., Kumkov S.S., Shilov E.A., Yurtaev D.A., Zenkov A.I. Specialized scientific visualization systems for optimal control application // Nonsmooth and Discontinuous Problems of Control & Optimization: Proc. IFAC Workshop Chelyabinsk, Russia, 17-20 June, 1998.- New York etc.: Pergamon, 1999.- P.71-76.
3. Averbukh V.L., Kumkov S.S., Pykhteev O.A., Yurtaev D.A. Visualization of level sets and singular surfaces in differential games // ALGORITHMY 2000: 15th Conf. on Sci. Computing, Vysoke Tatry-Podbanske, Slovakia, Sept. 10-15, 2000: Proc.- Bratislava, 2000.- P.196-206.