

СИСТЕМА OPENMODELER&VISUALIZER И ЕЕ ПРИЛОЖЕНИЯ В ВЫЧИСЛИТЕЛЬНОЙ МЕХАНИКЕ

В.А. Семенов, С.В. Морозов, О.А. Тарлапан

Институт системного программирования РАН, Москва, Россия

В статье рассматривается система визуализации общего назначения OpenModeler&Visualizer (OpenMV), позволяющая воспроизводить различные виды физических явлений посредством визуализации и анимации предварительно подготовленных данных моделирования в рамках произвольных сценариев, задаваемых пользователем. Открытая архитектура системы позволяет разрабатывать интегрированные программные комплексы для математического моделирования в различных областях науки и техники. Приводятся примеры использования системы для изучения ряда актуальных задач вычислительной механики. Работа поддержана грантами РФФИ (98-01-00321) и ИНТАС (96-0778).

THE SYSTEM OPENMODELER&VISUALIZER AND ITS APPLICATIONS IN COMPUTATIONAL MECHANICS

V.A. Semenov, S.V. Morozov, O.A. Tarlapan

Institute for System Programming RAS, Moscow, Russia

The general-purpose visualization system OpenModeler&Visualizer (OpenMV) allowing to reproduce different types of physical phenomena by means of visualization and animation of preliminary prepared modeling data in the scope of arbitrary user-defined scenarios is considered in the paper. Open architecture of the system allows to develop integrated systems for mathematical modeling in different scientific and technical areas. Examples of usage of the system for studying several actual problems of computational mechanics are presented. The work is supported by the grants of RFBR (98-01-00321) and INTAS (96-0778).

1. Введение

Создание современных сложных интегрированных распределенных приложений моделирования и визуализации связано с решением целого круга проблем концептуального, методологического и инструментального характера и обычно требует значительных усилий со стороны коллективов разработчиков. К подобным проблемам относятся, например, проектирование архитектуры приложений, интеграция функционально разнородных компонентов, организация дружественного интерфейса пользователя, обеспечение открытости, переносимости, эффективное использование на параллельных вычислительных системах.

Существенно упростить разработку программных приложений удастся в случае применения открытой модульной архитектуры в рамках общей инструментальной технологии. Представляемая система OpenMV (OpenModeler&Visualizer), предназначенная для создания сложных графических интерактивных вычислительных приложений математического моделирования и визуализации в различных областях науки и техники, является практическим воплощением такого подхода.

2. Организация и состав системы OpenMV

В соответствии с предложенной методологией разработки интегрированных приложений [1] на платформе UNIX/X Window была реализована система визуализации общего назначения OpenMV. Система позволяет пользователю воспроизводить различные виды физических явлений посредством визуализации и анимации предварительно подготовленных данных моделирования. Система обеспечивает визуализацию распространенных типов данных с использованием широкого набора методов в рамках произвольных сценариев, задаваемых пользователем.

Оригинальная открытая архитектура системы [2] позволяет на единой основе проводить ее функциональное наполнение и специализацию для конкретных предметных областей и прикладных проблем. Система включает в себя объектно-ориентированное ядро, инвариантное по отношению к различным задачам, и унифицированный пользовательский интерфейс. Создание

приложения на ее основе сводится к разработке библиотек классов для представления данных и алгоритмов, специфичных для рассматриваемой предметной области, и их регистрации в программном приложении. Состав прикладных библиотек и семантика входящих в них классов в конечном счете определяют функциональность всего приложения.

Объектно-ориентированное ядро является системой абстрактных базовых классов, выражающих высокоуровневые понятия объекта, научных данных, алгоритма, а также контейнерного класса для представления произвольных композиционных сцен. Итоговое изображение рассматривается как композиция связанных типизированных научных данных, а также алгоритмов моделирования и визуализации и представляет собой результат их последовательной активизации.

Базовым абстрактным классом ядра является класс *Object*, который выражает сущность произвольных данных и алгоритмов, участвующих в моделировании и визуализации. Объекты могут вводиться, конструироваться, связываться друг с другом, преобразовываться, запоминаться, визуализироваться, копироваться, передаваться, перераспределяться и уничтожаться в ходе работы приложения. Каждый конкретный объект $obj \in Object$ имеет свой собственный набор атрибутов, определяющий его внутреннее состояние и поведение, а также набор типизированных соединений. Соединения являются внешними портами объектов, посредством которых они могут быть связаны с другими подобными объектами. Тип конкретного соединения определяет потенциальную возможность связи данного объекта с любым другим объектом, тип которого удовлетворяет типу соединения или, другими словами, является его подтипом. Наличие связей в сцене характеризует функциональную зависимость ее объектов и, следовательно, необходимость их совместного рассмотрения и анализа. Каждая установленная связь определяет отношения использования главным объектом других вспомогательных объектов, с которыми он связан.

Основными понятиями ядра являются также объекты-данные и объекты-алгоритмы. Абстрактные классы *Data*, *Algorithm* являются производными от *Object* и наследуют его свойства и поведение. Класс *Data* выражает сущность разнообразных научных данных, возникающих в приложениях моделирования и визуализации. Объекты-данные являются пассивными объектами, контролирующими только собственное поведение. Класс *Algorithm* представляет различные алгоритмы, преобразования, операции и вспомогательные утилиты, реализующие все процессы в приложениях моделирования и визуализации. Алгоритмы являются активными объектами, контролирующими не только собственное поведение, но и поведение связанных с ними вспомогательных объектов (не обязательно данных). Отличительная особенность алгоритмов заключается в возможности их активизации, которая реализуется при наступлении соответствующих событий в сцене. В этом случае сцена активизирует соответствующий метод выполнения алгоритма. Предполагается, что к моменту инициации все связи алгоритмов были установлены. В противном случае алгоритм не активизируется и рассматривается как объект-данные.

Наконец, класс *Scene* является контейнерным классом, поддерживающим упорядоченное представление (композицию) всех объектов данных и алгоритмов, включенных в сцену, и обеспечивающим широкую функциональность, необходимую для разрабатываемых приложений. Сцена осуществляет регистрацию типов объектов, включаемых в нее, манипулирование ими, их ранжирование и упорядочивание, инициирование выполнения отдельных алгоритмов и всего сценария. Конструирование результирующей сцены осуществляется в результате непосредственного манипулирования ее объектами и активизации конвейера, составленного из ее отдельных алгоритмов. Алгоритмы конвейера сцены предварительно упорядочиваются и затем активизируются для решения частных задач моделирования и визуализации. В ходе выполнения алгоритмов объекты сцены взаимодействуют друг с другом, что приводит к созданию новых объектов и обновлению уже существующих. Располагая описанным представлением сцены, удается итеративно модифицировать ее путем редактирования объектов, уточнения состава и порядка применяемых методов моделирования и визуализации, корректировки их параметров вплоть до получения требуемой итоговой картины. Это же представление может использоваться для моделирования динамических процессов и анимации полученных результатов, поскольку однажды построенный сценарий может затем многократно применяться к семантически эквивалентным наборам данных, генерируемых внутри системы или вводимых извне.

Заметим, что для функционирования ядра не требуется какая-либо конкретизация семантики объектов, способов их представления и программной реализации. Лежащие в его основе механизмы связывания и взаимодействия объектов позволяют составлять и интерпретировать сложные сценарии, возникающие в реальных задачах, без какого-либо уточнения семантики конкретных объектов приложения. Следовательно, в большинстве случаев создание приложения сводится к разработке прикладной библиотеки классов для представления используемых научных данных и для реализации методов моделирования и визуализации, специфичных для конкретной предметной области. Принимая во внимание преимущества объектно-ориентированного программирования, реализацию прикладных библиотек классов данных и алгоритмов можно значительно упростить за счет неформального использования принципов инкапсуляции, наследования и полиморфизма. Таким образом, общность принятой системы абстракций, а также поддерживаемые ядром гибкие механизмы составления сцен и интерпретации определяемых ими сценариев обеспечивают существенную для приложений вычислительной механики возможность интеграции средств для описания геометрии расчетной области, задания начальных и граничных условий, моделирования физических процессов, численного решения типовых математических задач, визуализации полученных результатов и реалистичного отображения итоговых сцен с применением элементов виртуальной реальности.

Интерфейс системы предоставляет унифицированные диалоги для составления и интерпретации сложных сценариев, а также для редактирования объектов, окна просмотра трехмерной сцены, меню и панели инструментов. Для ясности и удобства использования диалоги сцены отображают иерархию зарегистрированных в системе прикладных классов, диаграмму текущего сценария и отфильтрованный список объектов сценария. Для создания, редактирования и связывания отдельного объекта внутри сценария пользователь устанавливает требуемые значения их открытых атрибутов и соединений через соответствующий унифицированный диалог редактирования. Унификация диалога достигается за счет применяемой спецификации атрибутов и соединений как членов классов данных и алгоритмов. Применяемая унификация интерфейса не препятствует его специализации для частных целей.

Через интерфейс пользователь выбирает соответствующий сценарий для изучаемой задачи, корректирует его параметры (уточняет параметры постановки задачи, настраивает используемые методы визуализации, устанавливает подходящие виды сцены, размещает источники света и т.д.), активизирует его и наблюдает возникающие и визуализируемые результаты как анимированное трехмерное изображение сцены. Итоговое изображение генерируется и анимируется посредством повторяющейся интерпретации выбранного сценария. Интерпретация может либо непосредственно инициироваться пользователем, либо осуществляться в результате некоторых изменений в сценарии, приводя к автоматическому обновлению итоговой сцены. Эта возможность позволяет интерактивно в визуальном режиме изучать решаемую задачу в зависимости от параметров ее постановки. Квалифицированные пользователи могут составлять собственные сценарии, интегрируя различные компоненты моделирования и визуализации для изучения сложных процессов и явлений.

Библиотека научных данных построена как набор классов, производных от ядра и реализующих различные научные понятия, обычно используемые в приложениях вычислительной механики. Библиотека включает классы для манипулирования различными типами данных, такими как структурированные и неструктурированные поверхностные и объемные сетки, ломанные, наборы точек, физические поля, цветовые палитры, шкалы, наборы маркеров, ортогональные сечения, виды. Библиотека организована как единая иерархия конкретных и абстрактных классов, наследуемых от класса *Data* ядра системы.

Используемые абстракции позволяют манипулировать унифицированным образом различными геометрическими и топологическими типами данных. Все классы геометрических данных имеют обобщенную реализацию, позволяющую использовать их как в двумерных, так и в трехмерных случаях и обеспечивающую их стандартное отображение в контексте OpenGL. Класс физических полей также поддерживает обобщенное многокомпонентное представление, которое может соответствовать скалярным, векторным и тензорным полям, заданным на различных типах многомерных геометрических данных.

Библиотека методов научной визуализации представляет собой набор производных от классов ядра алгоритмических классов, которые реализуют широко распространенные методы и способы, предназначенные для визуализации задач вычислительной механики. Библиотека включает классы для интерполяции полей, построения изолиний и изоповерхностей, конструирования линий и трубок тока, трассировки траекторий частиц, конструирования ортогональных сечений, маркеров и вычисления норм полей. Библиотека построена как единая иерархия конкретных и абстрактных классов, наследованных от класса *Algorithm* ядра.

Кроме методов визуализации, данная библиотека обеспечивает специальные классы для растеризации геометрических данных в графическом контексте OpenGL в соответствии с заданными правилами закраски или псевдозакраски, а также вспомогательные классы для сохранения/восстановления объектов сцены в файлы/из файлов внутреннего формата и сохранение созданных изображений сцены в растровом формате PPM.

Система OpenMV предоставляет также важные возможности для создания параллельных и распределенных приложений, необходимых при проведении расчетов на вычислительных системах с массовым параллелизмом, кластерах персональных компьютеров и рабочих станций, при проведении комплексных междисциплинарных исследований, обеспечении доступа к вычислительным центрам массового пользования и т.п. Детали разработки параллельных приложений на основе ядра OpenMV приводятся в [3].

Стандартное функциональное наполнение системы OpenMV разработанными библиотеками классов научных данных и методов визуализации обеспечивает:

- трехмерную визуализацию и анимацию распространенных типов геометрических и физических данных с применением широкого набора методик;
- интерактивное составление и интерпретацию сценариев с использованием парадигм объектно-ориентированного и визуального программирования;
- сохранение данных, сценариев, изображений в файлах внутреннего формата;
- эффективную обработку и визуализацию больших объемов данных с использованием современных параллельных технологий.

Использование в качестве базовых средств языка Си++, графического стандарта OpenGL, коммуникационного интерфейса MPI, стандартных библиотек пользовательского интерфейса Xlib и Motif позволяет относительно просто переносить систему на разные аппаратные платформы.

3. Примеры визуализации задач вычислительной механики в системе OpenMV

Рассматриваемая система OpenMV может применяться для изучения широкого класса задач вычислительной механики с использованием различных сценариев визуализации. Построение конкретного сценария осуществляется путем конструирования экземпляров соответствующих научных данных и алгоритмов визуализации и установления связей между ними в соответствии с общей дисциплиной связывания объектов. В определенном смысле получаемое представление сценария напоминает диаграмму потоков данных в системах визуализации общего назначения, таких как AVS, Data Explorer, IRIS Explorer [4], за исключением того, что в нем непосредственно участвуют и сами преобразуемые данные. Интерпретация сценария выполняется путем повторяющейся последовательной активизации алгоритмов, которые должны быть предварительно упорядочены в соответствии с установленными отношениями зависимости между объектами сцены. Будучи последовательно активизированы, алгоритмы сценария конструируют новые объекты и обновляют существующие, приводя к получению итогового изображения в статическом режиме и к его анимации в динамическом режиме. Однажды сконструированный сценарий может быть впоследствии снова и снова применен для изучения аналогичной задачи.

Необходимо отметить, что большинство методов визуализации допускает обобщенную реализацию, не зависящую от конкретных типов входных данных. Привлечение механизма абстрактных типов данных существенно облегчает построение сложных композиционных сценариев с использованием относительно небольшого репертуара методов. Данное обстоятельство оказывается существенным для функционального расширения открытой системы, поскольку добавление новых производных типов данных и алгоритмов не приводит к необходимости создания новых версий ранее реализованных классов.

Разработанная система визуализации была успешно использована для изучения актуальных научных проблем, связанных с математическим моделированием сложных физических явлений, таких как:

- обтекание открытой каверны набегающим потоком сверхзвукового газа;
- экологические последствия процесса горения метана;
- моделирование процессов переключения в полупроводниковых кристаллах с оптической бистабильностью.

Более подробную информацию о данных приложениях, дополнительные изображения и фильмы, описания других задач, изученных посредством OpenMV, можно найти в [5]. Строгая математическая постановка вышеперечисленных задач и детали численного моделирования приводятся в [6, 7].

Рассмотрим пример визуализации результатов моделирования сверхзвукового потока вязкого газа над открытой прямоугольной каверной. Данная задача является важной для проектирования и конструирования авиационной и ракетной техники. Для получения полной картины возвратного течения вблизи угла каверны применяется метод конструирования линий тока. Линии тока окрашиваются в соответствии с амплитудой скорости. Для информативности дополнительно визуализируется поле давлений с применением контурного метода. Полученное итоговое изображение показано на рисунке 1.

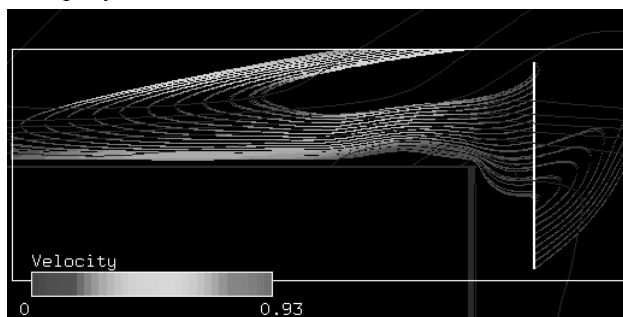


Рис. 1. Итоговое изображение для задачи динамики сверхзвукового газа

Другая задача связана с оценкой последствий процесса горения метана, имеющей важное экологическое значение. Фактор распространения потока воздуха является существенным для адекватного моделирования протекающих процессов химической кинетики для многокомпонентной реагирующей смеси. Рисунок 2 демонстрирует потоки газа, воспроизведенные посредством стрелок. Концентрация угарного газа, образующегося в результате сгорания метана, показана с помощью изолиний. Полученные изолинии и стрелки изображаются на фоне, окрашенном в соответствии со значениями температуры внутри факела.

Третья задача, связанная с моделированием процессов переключения в полупроводниковых кристаллах с оптической бистабильностью, является важной для конструирования новых полупроводниковых устройств, являющихся базой для будущих оптических ЭВМ и сетей. Процесс перехода распределения носителей заряда в оптических кристаллах в устойчивое состояние исследуется в зависимости от интенсивности лазерного излучения и диффузии носителей. На рисунке 3 поле концентрации носителей заряда визуализировано методом псевдозакраски на ортогональных сечениях кристалла.

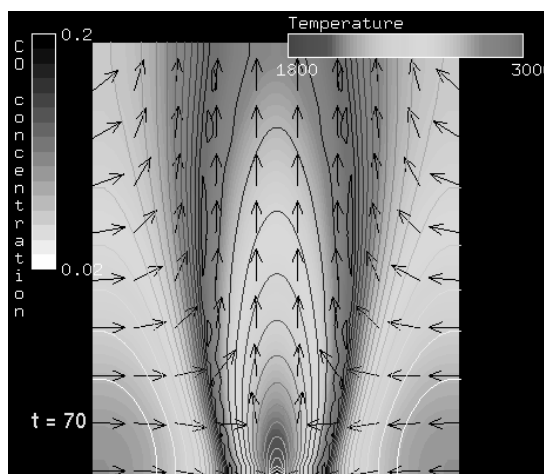


Рис. 2. Итоговое изображение для задачи химической кинетики

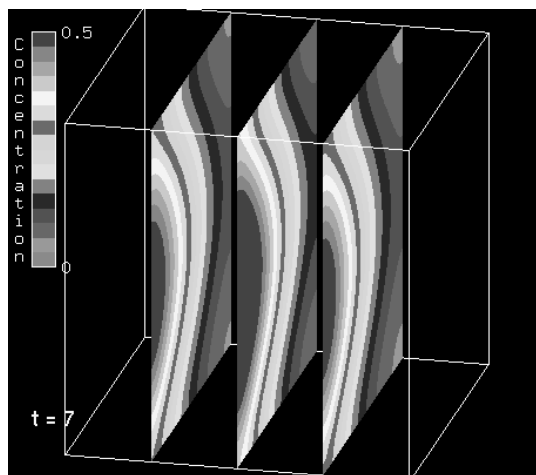


Рис. 3. Итоговое изображение для задачи оптики

4. Заключение

Таким образом, разработанная система OpenMV предоставляет развитые графические интерактивные возможности для визуализации и анимации распространенных научных данных с использованием широкого набора методов визуализации в рамках произвольных сценариев, определяемых пользователем. При этом открытая архитектура системы позволяет разрабатывать сложные интегрированные программные комплексы моделирования для изучения разнообразных физических явлений и технологических процессов. Ввиду полной инвариантности ядра системы, расширяемости библиотек прикладных классов и унифицированности графического интерфейса пользователя разработка подобных комплексов может осуществляться в рамках единой технологии на основе общих инструментальных средств широким кругом прикладных программистов.

В связи с этим открытая система OpenMV представляется перспективной для создания сложных графических приложений, интегрированных CAD/CAM/CAE систем в существенно разных предметных областях науки и техники на единой методологической, инструментальной и программной основе.

1. В.А. Семенов, П.Б. Крылов, С.В. Морозов, М.Г. Роминов, О.А. Тарлапан. Объектно-ориентированная методология разработки интегрированных приложений моделирования и визуализации // Труды Института системного программирования РАН. — М.: НСК РАН, 1999 (в печати).
2. В.А. Семенов, П.Б. Крылов, С.В. Морозов, О.А. Тарлапан. Объектно-ориентированная архитектура для приложений научной визуализации и математического моделирования // Программирование, 1999, № 5 (в печати).
3. V. Ivannikov, S. Morozov, V. Semenov, O. Tarlapan, R. Rasche, T. Jung. Parallel Object-Oriented Modeling and Visualization in OpenMV Environment // Proceedings of Graphicon'99, Moscow, 26 Aug.–1 Sep. 1999 (accepted).
4. S. Belien et al. Comparison of Visualization Techniques and Packages with applications to plasma physics, <http://home.sara.nl/~rik/REPORT.update>
5. The ISP RAS Scientific Visualization Group Home Page, <http://www.ispras.ru/~3D>
6. B.N. Chetverushkin, M.V. Iakovovski, M.A. Kornilina, K.Yu. Malikov, N.Yu. Romanukha. Ecological after-effects numerical modelling under methane combustion // Mathematical Models of Non-Linear Excitations, Transfer, Dynamics, and Control in Condensed Systems and Other Media. — Plenum Press, New York, 1998.
7. Yu.N. Karamzin, T.A. Kudryashova, S.V. Polyakov, I.G. Zakharova. Simulation of 3D optical bistability problem on parallel computer systems // Fundamental physical and mathematical problems and modelling of technical and technological systems. — MSTU Stankin, Moscow, 1999, pp. 117–124.