



Ядерные технологии в САПР

Опыт интеграции ядра С3D в САПР «БАЗИС»

Роман Колесников

Ядро геометрического моделирования — сложный программный продукт, при интеграции которого с прикладными программами неизбежны различные проблемы. В статье рассмотрен опыт интеграции ядра С3D с САПР корпусной мебели БАЗИС.

Если рынок «больших» программ проектирования давно поделен между несколькими крупными



Роман Колесников, программист
ООО «Базис-Центр»

игроками, такими как AutoCAD, SolidWorks, NX, Creo Elements и CATIA, то рынок специализированных программ автоматизированного проектирования всего и вся — окон и лестниц, корпусной и мягкой мебели, трубопроводов и швейных изделий — весьма широк и динамичен. Основных причин для этого две:

- высокая стоимость покупки «большой» САПР и сотрудника, умеющего в ней эффективно работать;
- отсутствие адаптации для проектирования конкретных изделий в «большой» САПР приводит к тому, что скорость проектирования специализированных изделий в них низкая.

Возможности специализации САПР

Ответом на указанные выше проблемы является создание специализированных (нишевых) САПР. Для этого у программистов есть два пути. Первый — доработка универсальной базовой САПР с ис-

пользованием предоставляемых API, плагинов и всевозможных скриптов. Этот подход не всегда оправдан. Стоимость такой специализированной системы для пользователя будет слишком высокой, поскольку платить надо как за базовую САПР, так и за ее адаптацию. Квалификация специалиста для работы с таким «комбайном» тоже должна быть достаточно высокой, что потребует значительных затрат на его обучение и последующее удержание в компании.

Второй путь — создание новой системы «с нуля». Этот путь для программистов, безусловно, гораздо сложнее, поскольку требуется разработать полный функционал автоматизации проектирования в определенной предметной области. Тем не менее он оказывается значительно более дешевым и удобным в использова-

нии для конечного пользователя, на которого, собственно, система и рассчитана. Именно популярность у конечных пользователей определяет успех того или иного программного продукта.

Естественно, создание нишевой САПР является очень затратным по времени и финансам. В ходе реализации такого проекта требуется разработка сложных алгоритмов геометрического моделирования, как правило, трехмерных объектов. Однако существуют стандартные библиотеки (геометрические ядра), которые могут помочь в решении этой задачи. О них и пойдет речь в данной статье. Предваряя обзор существующих библиотек, необходимо определить функционал, который требуется от геометрического ядра для автоматизации проектирования корпусной мебели.

Особенности моделирования корпусной мебели

Прежде всего, определим сам объект проектирования. Корпус-

ная мебель — это изделия, корпус (внутренний объем) которых образуется сочленением вертикальных и горизонтальных поверхностей (панелей) — плоских или рельефных. Для соединения панелей между собой используется стандартная крепежная фурнитура, а для придания им требуемой функциональности и эстетичности — декоративная фурнитура. Изделия могут иметь различные типовые блоки, например двери (фасады) или выдвижные ящики.

В данной статье речь пойдет о системе БАЗИС [1, 2, 3]. Это комплексная САПР, функционал которой охватывает все этапы проектирования, изготовления и реализации корпусной мебели. Исходя из особенностей моделирования мебельных изделий, требования к геометрическому ядру сформировались следующим образом: возможность моделирования плоских пластин произвольной формы с вырезами, профилей с постоянным сечением, тел вращения и так называемых гнутых панелей (рис. 1). Математически гнутая



Рис. 1. Гнутые панели

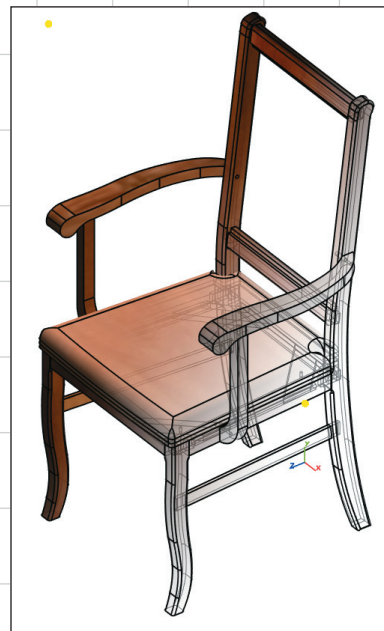


Рис. 2. Объекты моделирования мебельного изделия



ИНСТРУМЕНТЫ АРМ

панель моделируется с помощью операции деформации плоских пластин с нанесенными вырезами и рельефами. Помимо этого в ядре должны быть реализованы операции конструктивной геометрии над указанными телами и построение на основе созданных тел проекций и сечений (рис. 2).

Способы 3D-моделирования объектов

С прикладной точки зрения в CAD все решения геометрического моделирования можно разделить на три группы, использующие полигональное, воксельное или граничное представление.

Полигональное моделирование объектов — это генерация объектов в виде набора полигонов, как правило треугольников, которые в дальнейшем визуализируются с помощью графических API, таких как OpenGL или DirectX. Этот способ наиболее прост в реализации. Создание полигональных тел и конструктивные операции с ними можно реализовать самостоятельно или воспользоваться открытыми библиотеками, например CGAL. При разработке системы БАЗИС использовался именно такой подход к моделированию. Было создано собственное геометрическое ядро, которое вполне себя оправдывало в течение долгих лет. При том что это была достаточно «дешевая» разработка, она, тем не менее, позволяла моделировать весьма сложные мебельные изделия, например гнутые фасады.

Однако полигональное представление имеет существенные недостатки, что, в итоге, и заставило искать альтернативные решения. Недостатки эти заключаются в следующем:

- в полигональных моделях нет информации об ограничивающих тела кривых (дуг, окружностей, сплайнов) на поверхностях, что существенно усложняет работу конструктора-мебельщика, поскольку затрудняет точные построения от неплоских поверхностей (цилиндрических, конусных, сферических, параболических и др.);
- низкое качество геометрии при CSG-операциях (операции конструктивной блочной геометрии);

- невозможность автоматического создания корректных чертежей спроектированных объектов, расчета точных площадей поверхностей и объемов тел, а также других характеристик, для которых требуется точная геометрия.

Воксельное моделирование также достаточно специфично и не предназначено для решения типичных задач автоматизированного проектирования.

Наиболее удачным для САПР решением является моделирование объектов с помощью граничного представления BREP, то есть представления трехмерных тел, которое описывает их границы в виде набора связанных граней, где каждая грань является контуром на поверхности (например, плоской, цилиндрической, сферической, конической или NURBS). Этот способ свободен от недостатков полигональных моделей, хотя использует их в качестве промежуточного представления для визуализации или расчетов. Именно он применяется в абсолютном большинстве CAD-систем. Однако математический аппарат для граничного моделирования несоизмеримо сложнее, а выбор библиотек весьма ограничен. Большинство из них написаны на языке C++, причем некоторые из них имеют обертки (интерфейсы) для использования в программах на других языках.

Ядра на основе граничного представления геометрии

Мировыми лидерами в данной области являются ядра Parasolid и ACIS. Они используются в подавляющем большинстве CAD. Наряду с ними известны CGM и Granite, компоненты САПР CATIA и Creo Elements, которые сравнительно недавно стали доступны для лицензирования. Все они обладают мощнейшей функциональностью, хорошо оттестированы и оптимизированы. Единственный их недостаток для отечественного разработчика — цена. Очень внушительные ежегодные выплаты вкуче с большими процентными отчислениями от каждой продажи созданного программного продукта делают разработку специализированных САПР-систем на

Рис. 1



МЫ ЗНАЕМ, КАК СДЕЛАТЬ МЕБЕЛЬНОЕ ПРОИЗВОДСТВО ЭФФЕКТИВНЫМ

- ПРОЕКТИРОВАНИЕ
- РАСКРОЙ МАТЕРИАЛОВ
- ПРИЕМ ЗАКАЗОВ
- ПРОИЗВОДСТВЕННАЯ ЛОГИСТИКА
- СТАНКИ С ЧПУ
- ЭКОНОМИКА
- ИНТЕГРАЦИЯ С ERP
- УДАЛЕННЫЙ ДОСТУП



+7(496) 623-09-90
WWW.BAZISSOFT.RU



их основе очень дорогостоящим мероприятием.

Существуют еще две коммерческие библиотеки — Solids++ и SMLib. Эти ядра не слишком хорошо известны. По крайней мере, не удалось найти информацию о полноценных CAD-системах, построенных на их основе. Согласно описаниям на сайтах разработчиков, обе они реализуют типичные для геометрических ядер операции. Насколько стабильна и надежна их работа — вопрос открытый. SMLib обладает весьма весомым преимуществом: в отличие от всех остальных коммерческих библиотек, она продается вместе с исходными текстами.

В период анализа рынка геометрических ядер стала доступной для лицензирования отечественная разработка — ядро C3D, созданное компанией АСКОН для САПР КОМПАС 3D и других продуктов компании. В то время оно только выходило на рынок и заинтересовало специалистов сбаластированным соотношением цены и функционала. Практически одновременно с ним на горизонте также появилось ядро R6K — амбициозный отечественный проект, о котором тогда было несколько публикаций в прессе [4, 5, 6]. Тем не менее вот уже несколько лет никаких данных о нем получить невозможно даже на официальном сайте. Последняя известная публикация датирована 2013 годом. Сведения о реализации каких-либо проектов САПР на его основе обнаружить не удалось, а официальная почта на письма не отвечает.

Последнее рассматриваемое ядро — OpenCASCADE. Это единственное бесплатное решение, на основе которого создан известный пакет FreeCAD. Ядро, несомненно, обладает развитым функционалом и достаточно удобным API, но, к сожалению, имеет большое количество недоработок, которые при тестировании выразились в виде неправильной геометрии при операциях вычитания тел с совпадающими гранями и ошибок генерации проекций и силуэтов деталей. Однако не стоит забывать, что это открытый проект, в котором каждый при наличии возможностей может самостоятельно исправлять ошибки и выполнять доработки. Ядро OpenCASCADE можно ис-

пользовать в коммерческих продуктах совершенно бесплатно, но тем не менее, за дополнительные модули и техническую поддержку придется платить.

Проблема выбора

Полноценно сравнить вышеприведенные графические ядра, конечно, невозможно без кропотливой работы с каждым из них, однако примерное представление о функционале можно получить по CAD-системам, разработанным на их основе. Исходя из этого, а также из необходимости поддержания стабильной стоимости системы БАЗИС с новым ядром для конечных пользователей, для окончательного решения были отобраны бесплатное OpenCASCADE и отечественное C3D. В результате всестороннего анализа возможностей, нюансов лицензирования и возможных рисков выбор был сделан в пользу C3D по трем основным причинам:

- более высокая стабильность работы, которая крайне важна в силу необходимости корректно импортировать огромное количество моделей, созданных на собственном ядре;
- невозможность даже примерной оценки затрат времени, которое понадобится на доработку OpenCASCADE;
- наличие в ядре C3D импорта/экспорта DXF и стабильных алгоритмов генераций проекций и силуэтов.

Было принято решение о внедрении ядра от компании АСКОН в САПР «БАЗИС». Процесс перевода системы БАЗИС на C3D оказался достаточно продолжительным, но в настоящее время он завершен. Опыт этой работы автор и хочет поделиться с читателями.

C3D в САПР корпусной мебели

Ядро C3D представляет собой набор C++-классов, описывающих BREP-модели, вспомогательные классы и набор алгоритмов, оформленных в виде глобальных функций. На первом этапе внедрения надо было решить, как вообще подключить C++-ядро к программе, которая по историческим причинам написана на Delphi. Написание C-интерфейса для всех необходимых функций казалось утомительным и негиб-

ким подходом. Выход нашелся достаточно быстро — написать промежуточную «интерфейсную» DLL-библиотеку на C++ с применением чисто абстрактных классов, которые можно напрямую использовать в Delphi, поскольку они содержат внутри себя указатель на VMT (таблица виртуальных методов), формат которой, к нашему счастью, у компиляторов MSVC и Embarcadero полностью совпадает.

В результате была написана DLL-«прослойка», предоставляющая необходимый функционал в стиле объектно-ориентированного программирования и экспортирующая лишь одну функцию из DLL — получение основного интерфейса.

Следующий этап — построение всех необходимых для моделирования мебели тел в ядре C3D по исходным данным, их триангуляция и получение триангуляции обратно для визуализации. Решение этой задачи не вызвало особых осложнений.

Топология тел создавалась на основе функций построения ExtrusionSolid и EvolutionSolid, а их комбинация друг с другом путем вычитания, сложения и пересечения выполнялась с помощью функции BooleanResult. На выходе этих функций получались тела MbSolid, которые и содержат данные BREP: грани MbFace, ребра MbEdge и т.п.

Всё решалось замечательно, пока не началось тестирование на более-менее сложных моделях. Оказалось, что обработка огромного количества объектов без кэширования выполняется, мягко говоря, не очень быстро: скорость по сравнению с собственным ядром упала более чем в 10 раз, что абсолютно недопустимо для практической работы. Надо отдать должное специалистам компании АСКОН. Они оперативно помогли переписать функции построения наиболее распространенных тел «вручную». Это означает, что в системе БАЗИС некоторые тела стали строиться «самостоятельно» путем создания каждой грани тела, описания всех ребер и связывания их друг с другом. Таким образом, на выходе получалось корректное замкнутое тело.

Следующим этапом была визуализация, для реализации которой требовалось «подхватить» триан-

гуляцию из C3D и «отрисовать» ее. Первоначально задача показалась простой с учетом имеющегося рендера на OpenGL. Однако обнаружилась неожиданная проблема. Для работы с моделями мебельных изделий крайне важна визуализация объектов с текстурами (рис. 3), а явной поддержки текстурных координат в ядре C3D нет. Причина заключается в том, что для машиностроения и строительства это, видимо, неактуально, поэтому в КОМПАС 3D их нет. Пришлось «выкручиваться» следующим образом: ядро C3D позволяет генерировать для треугольников так называемые параметрические координаты, и для каждой грани можно подсчитать коэффициенты масштабирования, которые позволят наложить текстуру более-менее реалистично. Однако этот способ не позволяет рассчитать текстурные координаты связано для нескольких граней. Пришлось написать дополнительные процедуры, чтобы с помощью атрибутов связывать друг с другом координаты текстуры на гранях. Кроме того, для визуализации и редактирования моделей нужно передавать различные данные, которые по рекомендациям специалистов АСКОН также сохранялись с помощью атрибутов, представляющих собой массивы типа «имя — значение», которые можно назначать на тела, грани и ребра.

Самой сложной частью интеграции стало моделирование тел гнутых фасадов. Эта задача выявила четыре главных недостатка ядра C3D с точки зрения работы с моделями мебельных изделий:

- отсутствие операций деформации;
- отсутствие прямого моделирования операции фрезерования, то есть перемещения 3D-тела фрезы по некоторой кривой;
- нестабильная работа конструктивных операций с телами, содержащими линейчатые поверхности;
- проблемы с точностью, усугубившиеся низкой точностью собственного ядра (точность проектирования мебели несравнима с точностью в машиностроении), модели от которого необходимо поддерживать, и отсутствием возможности управления допусками в ядре C3D.



Рис. 3. Модель с уникальной текстурой

Однако следует отметить, что эти недостатки указаны по отношению к возможностям ядер Parasolid и ACIS, стоимость которых несравнима со стоимостью СЗД. Специалисты компании АСКОН работают в этих направлениях. В системе БАЗИС указанные недостатки частично устранены «обходными путями», частично «заморожены» на какое-то время. На работе 99,9% конечных пользователей это никак не сказывается.

Завершающими этапами интеграции стали различные операции экспорта/импорта (STEP, DXF), анализа взаимного расположения тел и т.п. Никаких особых проблем при их реализации отмечено не было.

Отдельно хочется упомянуть об отладке ядра СЗД, что, по сути, и происходило при его интеграции с системой БАЗИС, поскольку «Базис-Центр» стал владельцем одной из первых лицензий. Естественно, при работе было выявлено

значительное количество нюансов, и все они очень оперативно решались разработчиками.

Заключение

Суммируем основные проблемы, которые были решены совместно со специалистами компании АСКОН в процессе перевода системы БАЗИС на ядро СЗД:

- недостаточно удобное и структурированное API с большим количеством легаси-кода (*англ. Legacy code* — унаследованный код, то есть код, который более не поддерживается и не обновляется, но используется);
- разделение всех типов на 2D/3D;
- отсутствие единообразия 2D-функций, «самописные» типы и структуры, повторяющие STL (библиотека стандартных шаблонов C++), но гораздо менее удобные;
- недокументированные механизмы формирования имен

всех элементов построения и необходимость их передачи даже в тех случаях, если они не используются;

- необходимость приведения всех данных к точности порядка 10^{-6} , отсутствие управления точностью и различные ошибки, происходящие при ее несоблюдении.

В целом ядра геометрического моделирования являются очень сложным программным обеспечением, поэтому разного рода проблемы при их интеграции просто неизбежны.

Для системы БАЗИС переход с внутреннего решения на полноценное ядро сэкономил огромное количество времени и решил целый пласт накопившихся проблем. Оценивая результаты проделанной работы и анализируя отзывы конечных пользователей, можно сделать однозначный вывод: ядро СЗД — весьма достойный продукт, который во многих случаях может стать отличной базой для разработки специализированных САПР.

Для системы БАЗИС можно выделить следующие основные плюсы от перехода на полноценное геометрическое ядро:

- экономия значительного объема времени, которое было бы потрачено на создание и поддержание собственного ядра;
- улучшение качества создаваемой геометрии, включая правильное построение криволинейных поверхностей и границ;
- получение качественных чертежей в автоматическом режиме с удалением невидимых линий;

- экспорт/импорт в «сапровские» 3D-форматы (STEP, SAT, XT, DXF). ▶

Литература

1. Бунаков П.Ю. БАЗИС 9. Новая версия — новый уровень 3D-моделирования // САПР и графика. 2014. № 4(210). с. 72-74.
2. Бунаков П.Ю. БАЗИС 8: инновационные технологии в проектировании мебели // САПР и графика. 2013. № 9(202). с. 74-77.
3. Бунаков П.Ю. Автоматизация проектирования корпусной мебели: основы, инструменты, практика / П.Ю. Бунаков, А.В. Стариков. М.: ДМК Пресс, 2009. 864 с., ил.
4. Баранов Л., Козлов С., Кураксин С., Снытников Н. Российское 3D-ядро «RGK» представлено международному САПР-сообществу [электронный ресурс] / http://isicad.ru/ru/articles.php?article_num=16168. Дата обращения — 3.09.2015 г.
5. Баранов Л., Козлов С., Сёмин Д., Снытников Н. Российское 3D-ядро. Часть I: архитектура и приложения [электронный ресурс] / http://isicad.ru/ru/articles.php?article_num=16129. Дата обращения — 3.09.2015 г.
6. Баранов Л., Козлов С., Сёмин Д., Снытников Н. Российское 3D-ядро. Часть II: функции и инструменты [электронный ресурс] / http://isicad.ru/ru/articles.php?article_num=16130. Дата обращения — 3.09.2015 г.