

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Металлургия и литейное производство»

А. В. Ткаченко, В. А. Жаранов

**САПР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ,
ОСНАСТКИ И ОБОРУДОВАНИЯ.
САПР МЕТАЛЛУРГИЧЕСКОГО
ПРОИЗВОДСТВА**

**КУРС ЛЕКЦИЙ
по одноименной дисциплине
для студентов специальностей
1-36 02 01 «Машины и технология литейного
производства» и 1-42 01 01 «Металлургическое
производство и материалобработка»
дневной и заочной форм обучения**

Электронный аналог печатного издания

Гомель 2011

УДК 621.745.5(075.8)
ББК 34.61я73
Т48

*Рекомендовано к изданию научно-методическим советом
механико-технологического факультета ГГТУ им. П. О. Сухого
(протокол № 5 от 15.06.2010 г.)*

Рецензент: зав. каф. «Инженерная графика» ГГТУ им. П. О. Сухого
канд. техн. наук, доц. *А. М. Селютин*

Ткаченко, А. В.
Т48 САПР технологических процессов, оснастки и оборудования. САПР металлургического производства : курс лекций по одноим. дисциплине для студентов специальностей 1-36 02 01 «Машины и технология литейного производства» и 1-42 01 01 «Металлургическое производство и материалобработка» днев. и заоч. форм обучения / А. В. Ткаченко, В. А. Жаранов. – Гомель : ГГТУ им. П. О. Сухого, 2011. – 125 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <http://lib.gstu.local>. – Загл. с титул. экрана.

ISBN 978-985-420-975-3.

Рассмотрены вопросы теории и практики использования технологий САПР в литейном производстве и металлургии.

Для студентов технических специальностей дневной и заочной форм обучения.

**УДК 621.745.5(075.8)
ББК 34.61я73**

ISBN 978-985-420-975-3

© Ткаченко А. В., Жаранов В. А., 2011
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2011

ВВЕДЕНИЕ

Современные требования к управлению качеством производства предполагают разумное и эффективное использование ресурсов при одновременном повышении конкурентоспособности выпускаемой продукции. Системы автоматизированного проектирования (САПР) являются важной составляющей производственного процесса. Они позволяют предприятиям эффективно внедрять инновационные технологии, резко сократить сроки вывода новых изделий на рынок, обеспечить индивидуализацию заказов, выполнять повышенные требования к качеству, оперативно получать достоверные данные о потребностях производства. Рост вычислительной мощности компьютеров и широкое распространение программного обеспечения проектирования и производства привели к тому, что инженеры могут использовать системы автоматизированного проектирования для решения повседневных задач.

Конкурентоспособность предприятия прямо пропорциональна уровню его информатизации, т. к. последняя определяет качество и быстроту взаимодействия с клиентами и партнерами, уровень управления ресурсами, финансами и подразделениями, уровень развития маркетинга, дизайна, конструирования, технологической подготовки производства, эффективность эксплуатации инженерной инфраструктуры территории и зданий.

Современные предприятия не смогут выжить во всемирной конкуренции, если не будут выпускать новые продукты лучшего качества, более низкой стоимости и за меньшее время. Поэтому они стремятся использовать огромные возможности памяти компьютеров, их высокое быстродействие и возможности удобного графического интерфейса для того, чтобы автоматизировать и связать друг с другом задачи проектирования и производства, которые раньше были весьма утомительными и совершенно не связанными друг с другом.

Потребностью сквозной интеграции процесса автоматизированного проектирования обусловлен всевозрастающий интерес к концепции PLM (Product Lifecycle Management – управление информацией о жизненном цикле изделия). PLM подразумевает организацию единого информационного пространства на производстве для инструментов САПР, средств проектирования процессов (CAPP) и управления данными (PDM). Внедрение современных программных продуктов позволяет объединить маркетинговые исследования, проектирование изделий, разработку технологических процессов и производство.

Практически, в случае широкого внедрения автоматизации, возникает целесообразность реорганизации самих процессов разработки и подготовки производства изделий. Речь идет о распараллеливании процессов (в том числе территориально, современные информационные технологии позволяют управлять этим), о модификации принципов взаимодействия конструкторских и технологических служб, об изменении принципов работы с нормативно-справочной информацией.

На протяжении многих лет одной из самых результативных мер для достижения высокой эффективности в области литейного производства и металлургии остается замена дорогостоящих и ресурсоемких опытных работ компьютерными экспериментами. Это обеспечивает экономию материалов, энергоносителей, рабочего времени, снижает расходы на амортизацию оборудования.

Только компьютерное проектирование технологии, совмещенное с имитационным моделированием, позволяет «заглянуть» внутрь изделия, увидеть характер протекающих в нем процессов, понять причины возникновения дефектов. Именно полнота получаемой информации делает компьютерные технологии самым быстрым способом достижения желаемого результата.

Системы анализа (Computer Aided Engineering – CAE или, по-русски, – СКМ) литейных процессов (ЛП) используются литейщиками всего мира как «виртуальный литейный цех», где можно реализовать и проверить любые идеи, возникающие в голове конструктора и технолога.

Для эффективной работы с существующим программным обеспечением и создания макросов и программ, автоматизирующих процесс проектирования, пользователь должен иметь представление не только о среде, в которой он работает, но и о принципах, лежащих в ее основе. Фундаментальное знание помогает студенту быстро изучить любую конкретную систему с конкретной средой и использовать ее максимально эффективно. Руководства пользователя, поставляемые с САПР, обычно уделяют основное внимание пользовательскому интерфейсу и синтаксису; предполагается, что пользователь имеет достаточную теоретическую подготовку. Без такой подготовки инженер встретит серьезные затруднения с терминологией системной документации, а еще большие сложности у него вызовет анализ сообщений об ошибках.

1. ОСНОВЫ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ. ОБЩИЕ ПОЛОЖЕНИЯ. КОМПОНЕНТЫ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

1.1. Введение в проектирование с использованием САПР

Создание нового технического объекта – сложный и длительный процесс, в котором стадия проектирования имеет решающее значение в осуществлении замысла и достижении высокого технического уровня.

Современная методология проектирования базируется на *системном подходе*. Технический объект при системном подходе рассматривается как сложная система, состоящая из взаимосвязанных, целенаправленно функционирующих элементов и находящаяся во взаимодействии с окружающей внешней средой. Это позволяет учесть все факторы, влияющие на его функционирование, и обеспечить создание технического объекта с высокими показателями эффективности и качества.

Одно из важнейших требований системного подхода заключается в необходимости рассматривать существование и функционирование технического объекта во времени и в пространстве. Описание существования объекта во времени приводит к понятию жизненного цикла, а в пространстве – к понятию внешней среды, с которой взаимодействует объект в процессе функционирования.

Жизненный цикл технического объекта представляет собой совокупность взаимосвязанных процессов создания и последовательного изменения его состояния от формирования исходных требований к объекту до окончания его эксплуатации. Жизненный цикл состоит из следующих стадий: *создание, производство, обращение и эксплуатация*. Каждая из стадий содержит целый ряд этапов, операций и процедур. Важно отметить, что все стадии жизненного цикла имеют прямые и обратные связи. Прямые связи очевидны. Так, качество проекта определяет надежность и эффективность технического объекта. Надежность сказывается на производственных и эксплуатационных издержках, а эффективность характеризует основные эксплуатационные свойства объекта (производительность, экономичность и др.). Но высокая эффективность новых разработок, в свою очередь, достижима лишь при учете результатов эксплуатации существующего технического объекта (или его аналога) и анализа технологических аспектов их производства. В этом случае имеют место обратные связи.

Сложность и взаимосвязанность процессов жизненного цикла требует глубокого и целенаправленного их изучения.

Процесс разработки начинается с запросов потребителей, которые обслуживаются отделом маркетинга, и заканчивается полным описанием продукта, обычно выполняемым в форме рисунка. Процесс производства начинается с технических требований и заканчивается поставкой готовых изделий.

Операции, относящиеся к процессу разработки, можно разделить на аналитические и синтетические. Как следует из рис. 1.1, такие первичные операции разработки, как определение необходимости разработки, формулирование технических требований, анализ осуществимости и сбор важной информации, а также концептуализация разработки, относятся к подпроцессу синтеза. Результатом подпроцесса синтеза является концептуальный проект предполагаемого продукта в форме эскиза или топологического чертежа, отражающего связи различных компонентов продукта. В этой части цикла делаются основные финансовые вложения, необходимые для реализации идеи продукта, а также определяется его функциональность. Большая часть информации, порождаемой и обрабатываемой в рамках подпроцесса синтеза, является качественной, а следовательно, неудобной для компьютерной обработки.

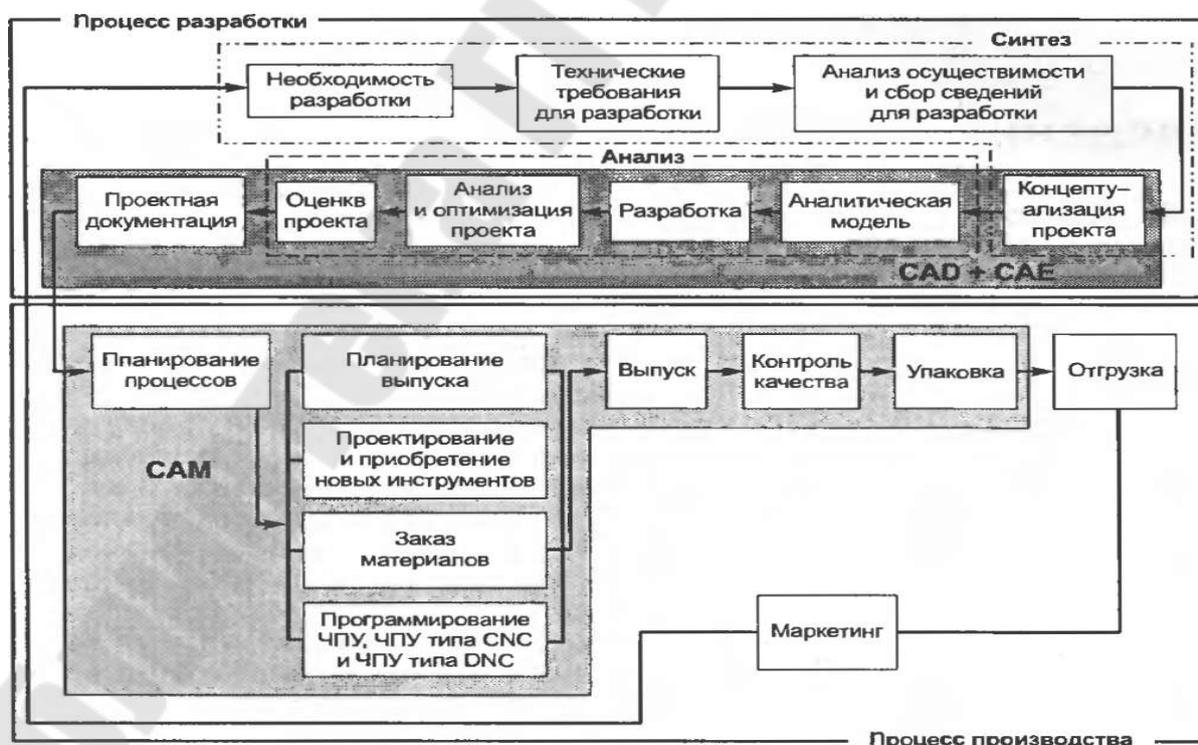


Рис. 1.1. Жизненный цикл продукта

Готовый концептуальный проект анализируется и оптимизируется – это уже подпроцесс анализа. Прежде всего вырабатывается аналитическая модель, поскольку анализируется именно модель, а не сам проект. Несмотря на быстрый рост количества и качества компьютеров, используемых в конструировании, в обозримом будущем отказаться от использования абстракции аналитической модели невозможно. Аналитическая модель получается, если из проекта удалить маловажные детали, редуцировать размерности и учесть имеющуюся симметрию. Редукция размерностей, например, подразумевает замену тонкого листа из какого-либо материала на эквивалентную плоскость с атрибутом толщины или длинного и тонкого участка на линию с определенными параметрами, характеризующими поперечное сечение. Симметричность геометрии тела и нагрузки, приложенной к нему, позволяет рассматривать в модели лишь часть этого тела. Типичные примеры анализа: анализ напряжений, позволяющий проверить прочность конструкции, контроль столкновений, позволяющий обнаружить возможность столкновений движущихся частей, составляющих механизм, а также кинематический анализ, показывающий, что проектируемое устройство будет совершать ожидаемые движения. Качество результатов, которые могут быть получены в результате анализа, непосредственно связано с качеством выбранной аналитической модели, которым оно ограничивается.

После завершения проектирования и выбора оптимальных параметров начинается этап оценки проекта. Для этой цели могут изготавливаться прототипы. В конструировании прототипов все большую популярность приобретает новая технология, названная *быстрым прототипированием* (rapid prototyping). Эта технология позволяет конструировать прототип снизу вверх, т. е. непосредственно из проекта, поскольку фактически требует только лишь данных о поперечном сечении конструкции. Если оценка проекта на основании прототипа показывает, что проект не удовлетворяет требованиям, описанный выше процесс разработки повторяется снова.

Если же результат оценки проекта оказывается удовлетворительным, начинается подготовка проектной документации. К ней относятся чертежи, отчеты и списки материалов. Чертежи обычно копируются, а копии передаются на производство.

Процесс создания разделяется на стадии: *предпроектные исследования, техническое задание, техническое предложение, эскизный проект, технический проект, рабочий проект, изготовление опыт-*

ных образцов, испытания и доводка, приемочные испытания. Первые две стадии и частично третья составляют этап внешнего проектирования, на котором осуществляется научно-технический поиск и прогнозирование, формирование описания среды функционирования технического объекта, моделирование и исследование, направленные на разработку концепции и технического решения. Этап внешнего проектирования, называемый также этапом научно-исследовательских работ (НИР), завершается разработкой технического задания.

Остальные стадии относятся к внутреннему проектированию и составляют этап опытно-конструкторских работ (ОКР), в процессе которого определяются и конкретизируются основные функциональные и конструктивные параметры, определяющие технико-экономические показатели и облик создаваемого технического объекта.

Решение проблемы создания нового технического объекта базируется на всесторонне обоснованной концепции и вытекает из безусловных потребностей общества, необходимости практической реализации достигнутого научного потенциала и повышения показателей эффективности.

Концепция определяется как комплекс требований к техническому объекту для выполнения его назначения и содержит описание основы функционирования объекта.

Кроме выделения стадий осуществляется *декомпозиция* процесса проектирования в зависимости от степени абстрагирования, характера отображаемых свойств объекта, его структуры, принятой схемы распределения работ между подразделениями проектно-конструкторской организации и др.

Декомпозиция приводит к выделению составных частей объекта (блоков), иерархических уровней, аспектов. Это позволяет сложную задачу проектирования свести к решению более простых задач с учетом взаимодействия между ними. Каждая задача решается на основе локальной оптимизации, но декомпозиция критериев при этом осуществляется таким образом, чтобы локальные цели были подчинены конечной цели проектирования. Следовательно, концепция системности выражается не только в выделении взаимозависимых и взаимодействующих элементов технического объекта как системы, но и в единстве целей их функционирования. Кроме того, технический объект, в свою очередь, рассматривается как элемент более сложной системы (надсистемы), в состав которой входит ряд объектов внешней среды, взаимодействующих с данным техническим объектом.

Таким образом, методология автоматизированного проектирования базируется на системном подходе, использующем принципы декомпозиции, иерархичности, итеративности, локальной оптимизации и комплексного осуществления процесса проектирования, включающего функциональный, конструкторский и технологический аспекты.

Аспекты различаются характером решаемых задач и используют различные описания.

Функциональный аспект включает отображение основных принципов функционирования, характера физических и информационных процессов в объекте. При функциональном проектировании осуществляется синтез структуры и определяются основные параметры объекта и его составных частей (элементов), оцениваются показатели эффективности и качества процессов функционирования. Результат проектирования – принципиальные, функциональные, кинематические, алгоритмические схемы и сопровождающие их документы.

Функциональное проектирование осуществляется практически на всех стадиях и этапах создания технического объекта и при этом многократно повторяется по мере раскрытия неопределенностей, характерных для начальных этапов.

Конструкторский аспект – это реализация результатов функционального проектирования. При конструкторском проектировании разрабатываются компоновки и рабочие чертежи деталей, осуществляется выбор стандартных и унифицированных элементов, материалов деталей, оформляется конструкторская и эксплуатационная документация.

При этом определяются оптимальные конструктивные параметры – размеры и форма деталей, сборочных единиц и т. п., обеспечивающие минимальные массу и габариты, равнопрочность элементов конструкции при заданном ресурсе.

Технологический аспект включает реализацию результатов конструкторского проектирования, т. е. их материализацию в виде физического изделия (машины, технической системы и т. п.). Технологическое проектирование решает задачи технологической подготовки производства. Разрабатываются технологические маршруты изготовления деталей, сборки, наладки и технологических испытаний изготавливаемых изделий, осуществляется выбор оборудования, оснастки, инструмента и т. д.

Кроме рассмотренной иерархии этапов, стадий и аспектов проектирования, иерархические уровни выделяют на основе блочного

структурирования технического объекта по функциональным признакам, а также в связи с различной степенью абстрагирования при описании физических свойств технического объекта на разных этапах и стадиях проектирования.

При блочном структурировании вначале выделяют крупные блоки, составляющие верхний иерархический уровень, затем каждый блок расчленяют на более мелкие блоки, входящие в следующий уровень, и т. д. вплоть до неделимых элементов (деталей), составляющих нижний уровень иерархии. Например, блоки верхнего иерархического уровня автомобиля: двигатель, трансмиссия, ходовая часть и др. В трансмиссию входят блоки: сцепление, коробка передач, карданная передача, главная передача, дифференциал. Каждый из них может быть, в свою очередь, расчленен на более мелкие блоки.

Как видно из рис. 1.1, процесс производства начинается с планирования, которое выполняется на основании полученных на этапе проектирования чертежей, а заканчивается готовым продуктом. Технологическая подготовка производства – это операция, устанавливающая список технологических процессов по изготовлению продукта и задающая их параметры. Одновременно выбирается оборудование, на котором будут производиться такие технологические операции, как получение детали нужной формы из заготовки. В результате подготовки производства составляются план выпуска, списки материалов и программы для оборудования. На этом же этапе обрабатываются прочие специфические требования, в частности рассматриваются конструкции зажимов и креплений. Подготовка занимает в процессе производства примерно такое же место, как подпроцесс синтеза в процессе проектирования, требуя значительного человеческого опыта и принятия качественных решений. Такая характеристика подразумевает сложность компьютеризации данного этапа. После завершения технологической подготовки начинается выпуск готового продукта и его проверка на соответствие требованиям. Детали, успешно проходящие контроль качества, собираются вместе, проходят тестирование функциональности, упаковываются, маркируются и отгружаются заказчику.

Выше был описан типичный жизненный цикл продукта. Рассмотрим теперь, каким образом на этапах этого цикла могут быть применены технологии CAD, CAM и CAE. Как уже говорилось, компьютеры не могут широко использоваться в подпроцессе синтеза, поскольку они не обладают способностью хорошо обрабатывать качественную информацию. Однако даже на этом этапе разработчик может,

например, при помощи коммерческих баз данных успешно собирать важную для анализа осуществимости информацию, а также пользоваться данными из каталогов.

Непросто представить себе использование компьютера и в процессе концептуализации проекта, потому что компьютер пока еще не стал мощным средством для интеллектуального творчества. На этом этапе компьютер может сделать свой вклад, обеспечивая эффективность создания различных концептуальных проектов. Полезными могут оказаться средства параметрического и геометрического моделирования, а также макропрограммы в *системах автоматизированной разработки чертежей* (computer-aided drafting). Все это типичные примеры систем CAD. Система геометрического моделирования (geometric modeling system) – это трехмерный эквивалент системы автоматизированной разработки чертежей, т. е. программный пакет, работающий с трехмерными, а не с плоскими объектами.

В аналитической фазе проектирования ценность компьютеров проявляется по-настоящему. Программных пакетов для анализа напряжений, контроля столкновений и кинематического анализа существует столько, что привести какие-либо названия смысла не имеет. Эти программные пакеты относятся к средствам автоматизированного конструирования (CAE). Главная проблема, связанная с их использованием, заключается в необходимости формирования аналитической модели. Проблемы не существовало бы вовсе, если бы аналитическая модель автоматически выводилась из концептуального проекта. Однако, как уже отмечалось, аналитическая модель не идентична концептуальному проекту – она выводится из него путем исключения несущественных деталей и редукции размерностей. Необходимый уровень абстракции зависит от типа анализа и желаемой точности решения. Следовательно, автоматизировать процесс абстрагирования достаточно сложно, поэтому аналитическую модель часто создают отдельно. Обычно абстрактная модель проекта создается в системе разработки рабочих чертежей или в системе геометрического моделирования, а иногда с помощью встроенных средств аналитического пакета. Аналитические пакеты обычно требуют, чтобы исследуемая структура была представлена в виде объединения связанных сеток, разделяющих объект на отдельные участки, удобные для компьютерной обработки. Если аналитический пакет может генерировать сетку автоматически, человеку остается задать только границы абстрактного объекта. В противном случае сетка также создается пользователем

либо в интерактивном режиме, либо автоматически, но в другой программе. Процесс создания сетки называется *моделированием методом конечных элементов* (finite-element modeling). Моделирование этим методом включает в себя также задание граничных условий и внешних нагрузок.

Подпроцесс анализа может выполняться в цикле оптимизации проекта по каким-либо параметрам. Разработано множество алгоритмов поиска оптимальных решений, а на их основе построены коммерчески доступные программы. Процедура оптимизации может считаться компонентом системы автоматизированного проектирования, но более естественно рассматривать эту процедуру отдельно.

Фаза оценки проекта также выигрывает от использования компьютера. Если для оценки проекта нужен прототип, мы можем быстро сконструировать его по заданному проекту при помощи программных пакетов, генерирующих код для машины быстрого прототипирования. Такие пакеты считаются программами для автоматизированной подготовки производства (САМ). Форма прототипа должна быть определена заранее в наборе входных данных.

Когда аналитические средства для работы с цифровыми копиями станут достаточно мощными, чтобы давать столь же точные результаты, что и эквивалентные эксперименты на реальных прототипах, цифровые копии начнут вытеснение обычных прототипов. Эта тенденция будет усиливаться по мере совершенствования технологий виртуальной реальности, позволяющих нам ощущать цифровую копию так же, как реальный прототип.

Последняя фаза процесса разработки – подготовка проектной документации. На этом этапе чрезвычайно полезным оказывается использование систем подготовки рабочих чертежей. Способность подобных систем работать с файлами позволяет систематизировать хранение и обеспечить удобство поиска документов.

Компьютерные технологии используются и на стадии производства. Процесс производства включает в себя планирование выпуска, проектирование и приобретение новых инструментов, заказ материалов, программирование машин с ЧПУ, контроль качества и упаковку. Компьютерные системы, используемые в этих операциях, могут быть классифицированы как системы автоматизированного производства. Например, программа автоматизированной технологической подготовки (computer-aided process planning – САРР) используется на этапе подготовки производства и относится к системам автоматизированного производства (САМ).

Подготовка производства с трудом поддается автоматизации, поэтому полностью автоматических систем технологической подготовки в настоящий момент не существует. Однако существует множество хороших программных пакетов, генерирующих код для станков с числовым программным управлением. Станки этого класса позволяют получить деталь нужной формы по данным, хранящимся в компьютере. Они аналогичны машинам для быстрого прототипирования. К системам автоматизированного производства относят также программные пакеты, управляющие движением роботов при сборке компонентов и перемещении их между операциями, а также пакеты, позволяющие программировать координатно-измерительную машину (coordinate measuring machine – СММ), используемую для проверки продукта.

1.2. Определение CAD, CAM, CAE

Согласно предыдущему разделу, *автоматизированное проектирование* (computer-aided design – CAD) представляет собой технологию, состоящую в использовании компьютерных систем для облегчения создания, изменения, анализа и оптимизации проектов. Таким образом, любая программа, работающая с компьютерной графикой, так же как и любое приложение, используемое в инженерных расчетах, относится к системам автоматизированного проектирования. Другими словами, множество средств CAD простирается от геометрических программ для работы с формами до специализированных приложений для анализа и оптимизации. Между этими крайностями уместятся программы для анализа допусков, расчета масс-инерционных свойств, моделирования методом конечных элементов и визуализации результатов анализа. Самая основная функция CAD – определение геометрии конструкции (детали механизма, архитектурные элементы, электронные схемы, планы зданий и т. п.), поскольку геометрия определяет все последующие этапы жизненного цикла продукта. Для этой цели обычно используются системы разработки рабочих чертежей и геометрического моделирования. Вот почему эти системы обычно и считаются системами автоматизированного проектирования. Более того, геометрия, определенная в этих системах, может использоваться в качестве основы для дальнейших операций в системах CAE и CAM. Это одно из наиболее значительных преимуществ CAD, позволяющее экономить время и сокращать количество ошибок, связанных с необходимостью определять геометрию

конструкции с нуля каждый раз, когда она требуется в расчетах. Можно, следовательно, утверждать, что системы автоматизированной разработки рабочих чертежей и системы геометрического моделирования являются наиболее важными компонентами автоматизированного проектирования.

Автоматизированное производство (computer-aided manufacturing – САМ) – это технология, состоящая в использовании компьютерных систем для планирования, управления и контроля операций производства через прямой или косвенный интерфейс с производственными ресурсами предприятия. Одним из наиболее зрелых подходов к автоматизации производства является числовое программное управление (ЧПУ, numerical control – NC). ЧПУ заключается в использовании запрограммированных команд для управления станком, который может шлифовать, резать, фрезеровать, штамповать, изгибать и иными способами превращать заготовки в готовые детали. В наше время компьютеры способны генерировать большие программы для станков с ЧПУ на основании геометрических параметров изделий из базы данных САД и дополнительных сведений, предоставляемых оператором. Исследования в этой области концентрируются на сокращении необходимости вмешательства оператора.

Еще одна важная функция систем автоматизированного производства – программирование роботов, которые могут работать на гибких автоматизированных участках, выбирая и устанавливая инструменты и обрабатываемые детали на станках с ЧПУ. Роботы могут также выполнять свои собственные задачи, например, заниматься сваркой, сборкой и переносом оборудования и деталей по цеху.

Планирование процессов также постепенно автоматизируется. План процессов может определять последовательность операций по изготовлению устройства от начала и до конца на всем необходимом оборудовании. Хотя полностью автоматизированное планирование процессов на данный момент практически невозможно, план обработки конкретной детали вполне может быть сформирован автоматически, если уже имеются планы обработки аналогичных деталей. Для этого была разработана технология группировки, позволяющая объединять схожие детали в семейства. Детали считаются подобными, если они имеют общие производственные особенности (гнезда, пазы, фаски, отверстия и т. д.) Для автоматического обнаружения схожести деталей необходимо, чтобы база данных САД содержала сведения о таких особенностях. Эта задача осуществляется при помощи объектно-ориентированного моделирования или распознавания элементов.

Дополнительно компьютер может использоваться для того, чтобы выявлять необходимость заказа исходных материалов и покупных деталей, а также определять их количество, исходя из графика производства. Называется такая деятельность планированием технических требований к материалу (*material requirements planning – MRP*). Компьютер может также использоваться для контроля состояния станков в цехе и отправки им соответствующих заданий.

Автоматизированное конструирование (computer-aided engineering – CAE) – это технология, состоящая в использовании компьютерных систем для анализа геометрии CAD, моделирования и изучения поведения продукта для усовершенствования и оптимизации его конструкции. Средства CAE могут осуществлять множество различных вариантов анализа. Программы для кинематических расчетов, например, способны определять траектории движения и скорости звеньев в механизмах. Программы динамического анализа с большими смещениями могут использоваться для определения нагрузок и смещений в сложных составных устройствах типа автомобилей. Программы верификации и анализа логики и синхронизации имитируют работу сложных электронных цепей.

Из всех методов компьютерного анализа наиболее широко в конструировании используется *метод конечных элементов (finite-element method – FEM)*. С его помощью рассчитываются напряжения, деформации, теплообмен, распределение магнитного поля, потоки жидкостей и другие задачи с непрерывными средами, решать которые каким-либо иным методом оказывается просто непрактично. В методе конечных элементов аналитическая модель структуры представляет собой соединение элементов, благодаря чему она разбивается на отдельные части, которые уже могут обрабатываться компьютером.

Для использования метода конечных элементов нужна абстрактная модель подходящего уровня, а не сама конструкция. Абстрактная модель отличается от конструкции тем, что она формируется путем исключения несущественных деталей и редуцирования размерностей. Например, трехмерный объект небольшой толщины может быть представлен в виде двумерной оболочки. Модель создается либо в интерактивном режиме, либо автоматически. Готовая абстрактная модель разбивается на конечные элементы, образующие аналитическую модель. Программные средства, позволяющие конструировать абстрактную модель и разбивать ее на конечные элементы, называются *препроцессорами (pre-processors)*. Проанализировав каждый элемент, компьютер

собирает результаты воедино и представляет их в визуальном формате. Например, области с высоким напряжением могут быть выделены красным цветом. Программные средства, обеспечивающие визуализацию, называются *постпроцессорами* (postprocessors).

Существует множество программных средств для оптимизации конструкций. Хотя средства оптимизации могут быть отнесены к классу CAE, обычно их рассматривают отдельно. Ведутся исследования возможности автоматического определения формы конструкции путем объединения оптимизации и анализа. В этих подходах исходная форма конструкции предполагается простой, как, например, у прямоугольного двумерного объекта, состоящего из небольших элементов различной плотности. Затем выполняется процедура оптимизации, позволяющая определить конкретные значения плотности, позволяющие достичь определенной цели с учетом ограничений на напряжения. Целью часто является минимизация веса. После определения оптимальных значений плотности рассчитывается оптимальная форма объекта. Она получается отбрасыванием элементов с низкими значениями плотности.

Замечательное достоинство методов анализа и оптимизации конструкций заключается в том, что они позволяют конструктору увидеть поведение конечного продукта и выявить возможные ошибки до создания и тестирования реальных прототипов, избежав определенных затрат. Поскольку стоимость конструирования на последних стадиях разработки и производства продукта экспоненциально возрастает, ранняя оптимизация и усовершенствование (возможны только благодаря аналитическим средствам CAE) окупаются значительным снижением сроков и стоимости разработки.

Таким образом, технологии CAD, CAM и CAE заключаются в автоматизации и повышении эффективности конкретных стадий жизненного цикла продукта. Развиваясь независимо, эти системы еще не до конца реализовали потенциал интеграции проектирования и производства. Для решения этой проблемы была предложена новая технология, получившая название *компьютеризованного интегрированного производства* (computer-integrated manufacturing – CIM). CIM пытается соединить «островки автоматизации» вместе и превратить их в бесперебойно и эффективно работающую систему. CIM подразумевает использование компьютерной базы данных для более эффективного управления всем предприятием, в частности бухгалтерией, планированием, доставкой и другими задачами, а не только

проектированием и производством, которые охватывались системами CAD, CAM и CAE. CIM часто называют философией бизнеса, а не компьютерной системой.

1.3. Структура и параметры объектов проектирования

Структура – это упорядоченное множество элементов и их отношений.

Технический объект при системном подходе рассматривается как система, состоящая из взаимодействующих элементов, составляющих упорядоченное множество.

Структура технического объекта характеризуется качественным и количественным составом элементов и их взаиморасположением или взаимосвязями. Качественное различие элементов определяется их физическими свойствами. Количественно физические свойства элементов выражаются некоторыми скалярными величинами, называемыми *параметрами элементов*.

Характеристики функционирования технического объекта зависят от его физических свойств и внешних воздействий окружающей среды.

Физические свойства объекта определяются его структурой и параметрами элементов, из которых он состоит. Внешние воздействия зависят от физических свойств внешней среды и характера ее взаимодействия с техническим объектом. Физические свойства внешней среды также определяются ее параметрами.

Параметр – это величина, характеризующая свойство или режим работы объекта. Под объектом здесь понимается как отдельный элемент технической системы, так и вся система в целом. Следует отметить, что параметрами технической системы являются *показатели качества и эффективности*: производительность, рабочая скорость, грузоподъемность, удельная материалоемкость, удельная энергоемкость, габариты, масса, показатели надежности, показатели качества переходных процессов и др. Эти параметры называют *выходными параметрами технического объекта*.

Если структура технического объекта определена, то его выходные параметры зависят только от параметров элементов и параметров внешней среды. Различают внутренние и внешние параметры.

Внутренние параметры – это параметры элементов, из которых состоит технический объект. Например, двигатель и трансмиссия яв-

ляются элементами автомобиля. Выходные параметры их – мощность двигателя, передаточные числа трансмиссии и одновременно это внутренние параметры автомобиля.

Выходные параметры характеризуют свойства технического объекта, а внутренние параметры – свойства его элементов.

При переходе к новому иерархическому уровню проектирования внутренние параметры могут стать выходными и наоборот.

Внешние параметры – это параметры внешней среды, оказывающей влияние на функционирование технического объекта.

1.4. САПР на базе Windows. Объектно-ориентированное программирование в САПР

Когда рынок программного обеспечения CAD/CAM/CAE достиг зрелости, ситуация на нем радикально изменилась. Инженеры и производители свыклись с тем, что им нужно нечто большее, нежели просто средства построения двумерных чертежей. Они давно ждали появления гибких систем построения трехмерных чертежей, которые могли бы использоваться на всех этапах: от проектирования до производства. До недавних пор промышленные приложения доминировали на рынке традиционных средств САД высшего класса. К счастью, персональные компьютеры со временем стали невероятно быстрыми и мощными, а многие разработчики программного обеспечения начали выпускать хорошие программные продукты, использующие преимущества превосходной графической среды Microsoft Windows. Первые программные продукты этой категории были выпущены в 1995 г., а первые версии большинства продуктов датированы 1996 г.

Все новые продукты обладали следующими общими особенностями. Во-первых, они разрабатывались с максимальным использованием функций Windows, и потому их интерфейсы получались схожими с интерфейсами других программ Microsoft. Фактически эти программы были похожи на другие продукты Microsoft для автоматизации конторского труда, поэтому пользователи чувствовали себя комфортно в знакомой среде и быстро осваивали их. Кроме того, новые программы поддерживали функции внедрения и связывания объектов (Object Linking and Embedding – OLE), характерные для офисных пакетов Microsoft. Таким образом, любое изображение трехмерной детали или устройства, созданное в пакете геометрического

моделирования, может использоваться другими программами Microsoft. Представьте, насколько удобна эта возможность при подготовке пользовательской документации. Вы просто копируете трехмерное представление деталей из системы CAD и вставляете его в нужное место текстового файла, созданного и редактируемого в текстовом процессоре. В настоящее время эта функция OLE расширяется с целью включения поддержки трехмерных данных, что значительно облегчит передачу данных между разными системами CAD.

Во-вторых, в новых системах использовался компонентный подход, согласно которому важнейшие компоненты программного обеспечения выбираются из доступных программ, после чего разработчик системы просто объединяет проверенные технологии, сосредоточивая свое внимание на деталях, относящихся непосредственно к проектированию. Например, разработчик системы может использовать ACIS от Spatial Technology, Parasolids от Unigraphics Solutions или Designbase от Ricoh в качестве моделирующего ядра (ключевого компонента, обеспечивающего работу с пространственными объектами в трех измерениях и Constraint Solver от D-cubed для параметрического проектирования). Такой подход позволяет сократить время на разработку, причем пользователю будут предоставлены именно те возможности, которые понадобятся ему в процессе проектирования конкретного продукта.

В-третьих, новые системы основаны на объектно-ориентированной технологии, три аспекта которой необходимо рассмотреть особо. С точки зрения программирования, объектно-ориентированная технология означает написание модульных программ таким образом, чтобы обеспечить независимое повторное использование модулей. Типичным объектно-ориентированным языком программирования является C++. Каждая функция может быть написана на этом языке таким образом, что она будет действовать как независимое целое. Объектно-ориентированная технология определяет также интерфейс между системой и пользователем. Объектная ориентированность пользовательского интерфейса означает, что каждый элемент интерфейса самостоятельно реагирует на изменения в ситуации на действия пользователя. Это значительно облегчает работу пользователя системой. Объектно-ориентированная технология используется и для эффективного хранения данных. В обычных системах CAD данные о детали обычно хранятся в нескольких файлах: один файл используется для геометрической формы, другой – для сетки конечных элементов, третий – для траектории движения фрезы станка с ЧПУ, и т. д. В объ-

ектно-ориентированных системах все данные, относящиеся к одной детали, хранятся в одном файле. При сохранении одинаковых данных в разных файлах происходит избыточное дублирование, а в объектно-ориентированных системах этого удастся избежать, что приводит к значительной экономии памяти.

В-четвертых, системы поддерживают либо параметрическое, либо вариационное моделирование. Оба подхода позволяют пользователю определять форму, задавая ограничения, а не характеристики отдельных элементов этой формы. Единственное различие в том, что в одном случае ограничения учитываются одновременно, а в другом – последовательно. Примером непосредственной работы элементами формы является определение прямоугольника как двух наборов параллельных отрезков, находящихся на конкретном расстоянии друг от друга. Однако тот же прямоугольник может быть определен при помощи ограничений, например, заданием условия перпендикулярности смежных отрезков и расстояния между параллельными отрезками. Многие системы, поддерживающие возможность параметрического или вариационного моделирования, воспринимают очевидные ограничения, такие как перпендикулярность и параллельность, непосредственно из начального эскиза пользователя, позволяя уменьшить объем вводимых данных. В этом случае от пользователя требуется только ввести размер, после чего он сможет изменять форму, варьируя эти размеры. Такая функция системы называется моделированием по размерам (*dimension-driven modeling*). Это означает, что геометрия определяется размерами детали и может быть легко изменена переопределением этих размеров. В таких системах пользователь может также указывать отношения между размерами, которые будут сохраняться при изменении самих размеров. Это один из механизмов, позволяющих сохранить идеи разработчика в конечном продукте. Однако задать все ограничения, определяющие геометрию, может быть достаточно затруднительно, особенно для сложной детали. В таких ситуациях системам, поддерживающим параметрическое и вариационное моделирование, требуются дополнительные сведения для проектирования.

Наконец, в системы встраивается поддержка совместного проектирования через Интернет. Эта поддержка позволяет удаленным пользователям работать над одной и той же деталью, имея перед глазами ее модель на своих экранах. Разработчики могут также проверять проект в целом, сравнивая свои детали с деталями других разработчиков. Для того чтобы воспользоваться этой возможностью, нужно как минимум описать деталь в формате VRML.

2. СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ЛИТЕЙНОГО ПРОИЗВОДСТВА

2.1. Автоматизация процесса разработки технологии получения отливок

Одним из важнейших направлений современного литейного производства является использование компьютерных технологий для решения разнообразных задач по проектированию технологической оснастки. В настоящее время на многих предприятиях используются компьютерные методы для построения чертежей на плоскости с использованием средств AutoCAD. Однако построение твердотельных 3D-моделей, являющееся необходимым этапом моделирования технологических процессов, еще не стало традиционным для значительной части конструкторских отделов промышленных предприятий.

Анализ функционирования конструкторских отделов показывает, что многие специалисты приобретают опыт построения твердотельных моделей на основе метода проб и ошибок. В этом случае технологии построения 3D-моделей не являются оптимальными и в зависимости от уровня компьютерной подготовки конструктора могут быть очень трудоемкими по времени. Этот фактор повышает время, затрачиваемое на процесс проектирования. Следует отметить, что при построении твердотельных моделей необходимо учитывать и то, что она разрабатывается с выбранной литниковой системой для моделирования процесса заполнения форм. Причем сама литниковая система может быть в пространственном отношении довольно сложным объектом. То есть должны быть «зарезервированы» некоторые варианты проектирования литниковой системы. Другими словами, при необходимости должна быть предусмотрена варьированность пространственной конфигурации литниковой системы для каждой отливки. Поэтому разработка алгоритмических схем автоматизации проектно-конструкторских работ, направленных на совершенствование технологического процесса, является актуальной научно-технической задачей САПР литейного производства.

На рис. 2.1 представлена обобщенная схема автоматизации процесса разработки технологии получения отливок литьем в песчано-глинистые формы. Как видно из рис. 2.1, процесс разработки может быть разбит на пять этапов. На первом этапе средствами твердотельного моделирования и чертежей заказчика строится 3D-модель отлив-

ки. Этот этап может быть автоматизирован по средствам выбора оптимальных алгоритмов с учетом разбиения отливки на выбранные примитивы.

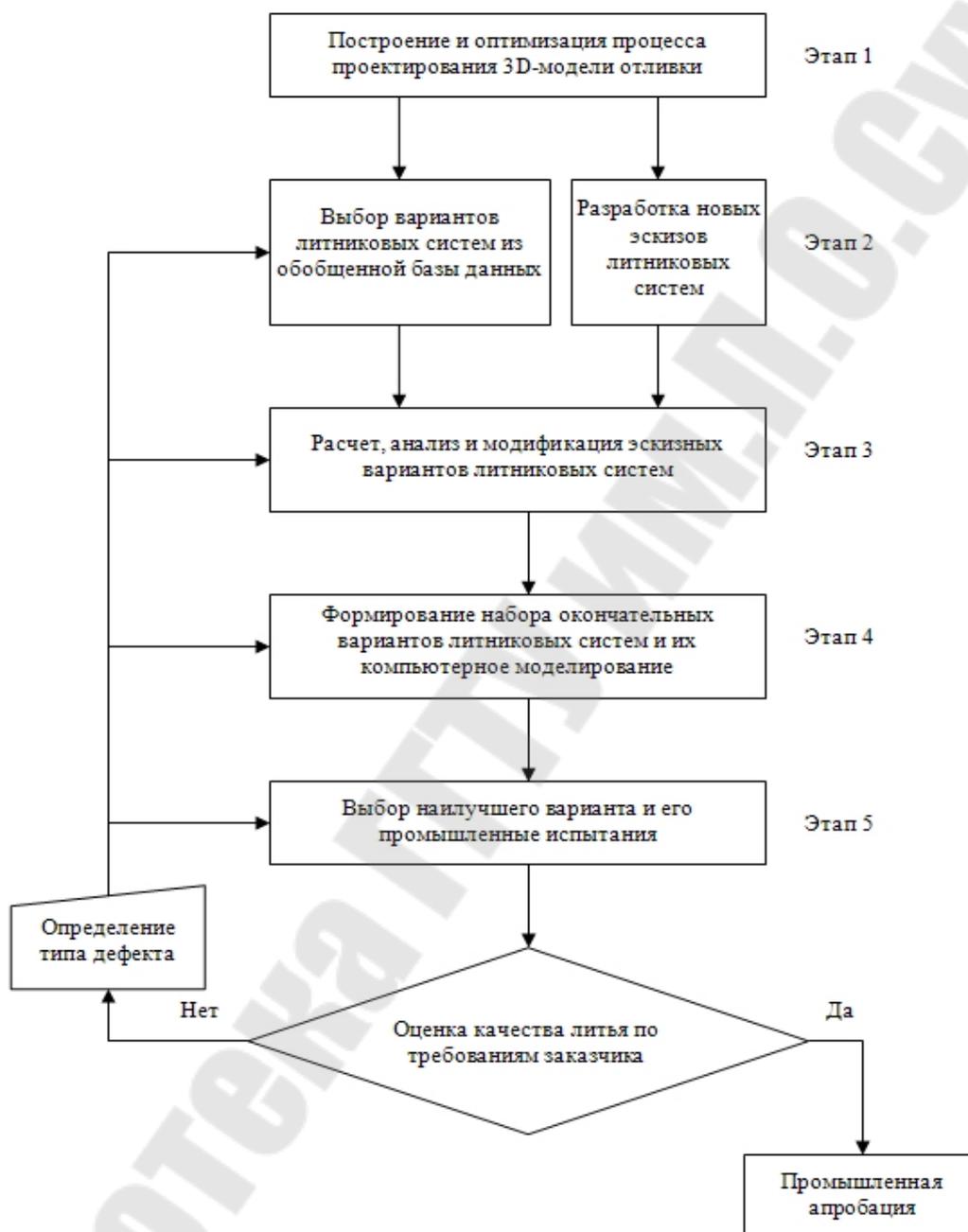


Рис. 2.1. Обобщенная схема автоматизации процесса разработки технологии получения отливок литьем в песчано-глинистые формы

Причем уже на этом этапе возможно использование методов классификации, позволяющих на ранних стадиях проектирования отнести деталь к некоторому классу, для которого ранее уже была разработана оптимальная цепочка проектирования примитивов. Цепочка

проектирования – это линейка шагов, последовательное выполнение которых позволит за минимально короткое время спроектировать 3D-геометрию отливки, относящейся к данному классу, с минимальным набором примитивов. При этом созданная 3D-геометрия будет гибка к внесению изменений в конструкцию отливки, и обладать широкой базой для проектирования к ней большого числа вариантов литниковых систем. На втором этапе для построенной 3D-модели необходимо выбрать литниковую систему, причем, с одной стороны, размер литниковой системы (объем, расход металла) должен быть минимальным в расчете на отливку. С другой стороны, литниковая система должна обеспечить ламинарное заполнение формы с отсутствием дефектов типа недолив при заданных температурах заливки используемого сплава. Для реализации этого этапа возможно как минимум два пути. Первый путь использование экспертной системы с базой данных по готовым техническим решениям, которые уже апробированы в промышленных условиях. То есть это прототипирование, основанное на адекватном (математическом) сопоставлении конструкторско-технологических и геометрических параметров, а также массовых характеристиках нового изделия, с соответствующими параметрами деталей, составляющих обобщенную базу предприятия.

В качестве методик сопоставления используются алгоритмы кластерного анализа, а также алгоритмы, построенные на идеологии нейронных сетей. В результате сравнения определяется класс, подкласс детали, а также предлагается ряд литниковых систем, используемых для наиболее близких, по выделенным признакам, деталей данного подкласса, хранящихся в обобщенной базе. Второй путь основывается на выборе литниковой системы при отсутствии аналога в базе данных. То есть конструктор при разработке литниковой системы пользуется своей алгоритмической схемой. Используя накопленный опыт, конструктор проектирует ряд принципиально отличных литниковых систем, которые могут быть применены к проектируемой отливке.

Выбранные на первом пути либо разработанные на втором литниковые системы не всегда являются оптимальными для данной детали, поэтому целесообразно использование их только в качестве начальных вариантов, которые в дальнейшем будут дорабатываться и модифицироваться.

Предлагаемые решения могут быть оценены на основе как инженерных методик, основанных на теплофизических расчетах, так и

методов компьютерного моделирования процесса заполнения формы. На третьем этапе отобранные на предыдущих стадиях эскизные варианты литниковых систем проходят ступени расчета, модификации и оценки.

Известно, что при разработке литниковой системы для изготовления отливки большое внимание уделяется соотношению сечений ее элементов. Однако, используя современные системы математического моделирования технологических процессов, можно убедиться, что классический расчет Озанна-Диттерта не всегда справедлив. Зачастую, сечения, рассчитанные по методике Озанна-Диттерта, не являются оптимальными для данной отливки, что в дальнейшем при использовании разработанной литниковой системы приводит к большому количеству бракованных отливок. Дальнейшее изменение и подбор приемлемых соотношений элементов литниковой системы становится затруднительным и носит скорее интуитивный характер. Как показывают практические данные, важность выбора соотношения элементов литниковой системы значительно преувеличена. Сечения элементов литниковой системы определяют лишь динамику заполнения полости формы и отвечают за образование лишь малой доли дефектов по недоливам и газовым включениям. В реальных условиях за возникновение большей части дефектов отливки отвечает положение мест подвода металла к отливке, траектория входа струи металла в полость формы и геометрия литниковой системы, не связанная с ее сечениями. Процесс адаптации, анализа и модификации литниковой системы идет по алгоритмической схеме, представленной на рис. 2.2.

Первоначально для каждого входного (базового) варианта литниковой системы по классической методике рассчитываются сечения ее элементов, с помощью предварительного компьютерного моделирования оценивается вероятность получения качественной отливки (оцениваются дефекты типа недолив).

На втором шаге рассматриваются различные варианты подвода металла к отливке, анализируется вероятность образования дефектов типа горячие трещины, усадочная пористость.

На третьем шаге рассматриваются варианты траектории подвода металла к полости формы, анализируются дефекты типа размыв формы.

На четвертом шаге рассматривается конструкция литниковых каналов. Протяженность литниковых каналов, с одной стороны, должна быть минимальна для уменьшения массы литниковой системы, с другой стороны, должна обеспечивать ламинарное заполнение полос-

ти формы. На четвертом этапе из вариантов и подвариантов эскизов литниковых систем, отобранных на третьем этапе, формируется набор литниковых систем, не превышающий 10 (цифра условна), которые должны пройти виртуальную апробацию с точки зрения математической логики моделирования процесса заполнения и кристаллизации сплава в форме. При этом одним из критериев выбора лучшего варианта должно быть отсутствие дефектов для нескольких уровней варьирования технологических факторов. То есть необходимо учесть трудности контроля технологических факторов, а также специфику нагревательных устройств, служащих для приготовления металла, и возможность поддержания необходимой температуры металла до его заливки в форму.

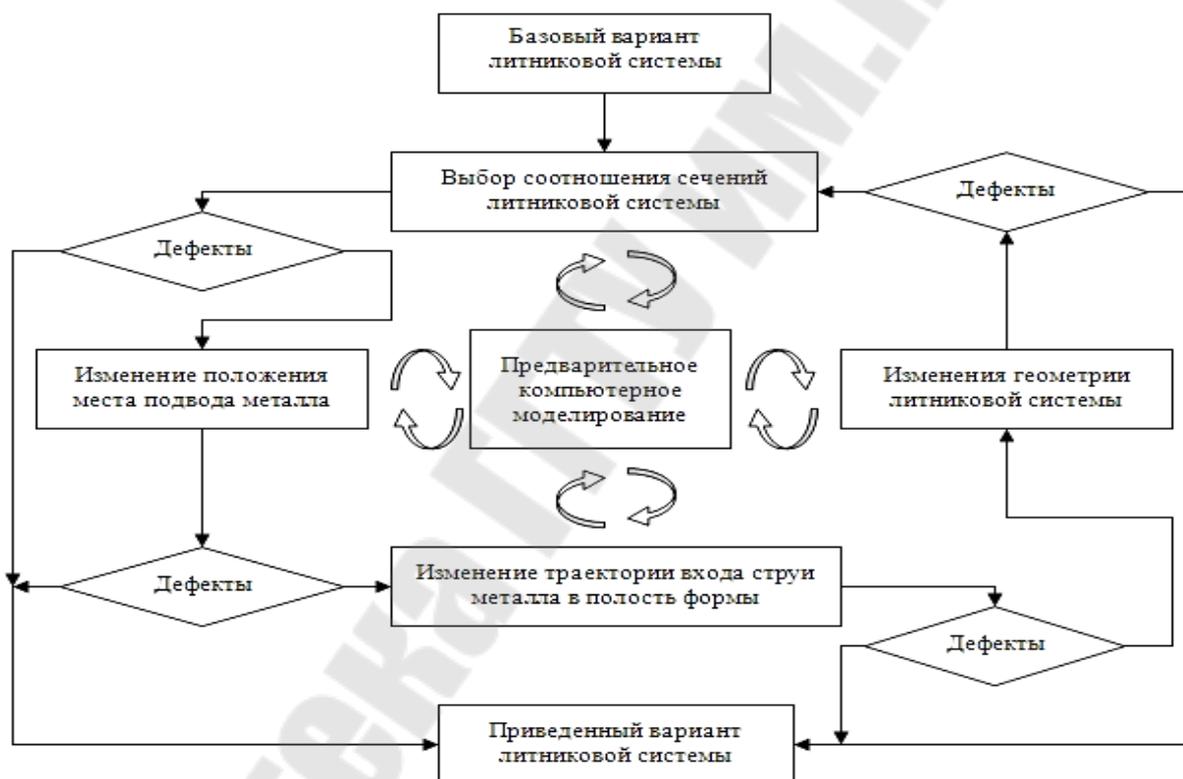


Рис. 2.2. Алгоритм оптимизации варианта литниковой системы

На пятом этапе проводится выбор наилучшей комбинации «отливка – литниковая система», для которой и проводятся промышленные испытания. В зависимости от результатов промышленных испытаний и анализа дефектов отливки принимается решение о том, к какому этапу проектирования необходимо перейти, чтобы обеспечить требуемый уровень свойств отливки.

2.2. Применение систем автоматизированного моделирования литейных процессов в опытном производстве литых заготовок

В современном литейном производстве, где разнообразие и сложность вопросов, связанных с проектированием и производством качественных отливок заданной точности очень велики, становится все более актуальным применение систем автоматизации процесса проектирования и моделирования литейных процессов. Далее описан алгоритм работы в СКМ ЛП «Полигон», разработанная в Центральном научно-исследовательском институте материалов (ЦНИИМ, г. Санкт-Петербург) и с системой ProCAST (USA). Они позволяют на основе моделирования тепловых условий затвердевания отливки оценить качество литниково-питающей системы (ЛПС) и температурно-временные параметры процессов литья не на реальной отливке, а на ее компьютерной модели.

При помощи СКМ ЛП «Полигон» и ProCAST можно моделировать следующие литейные процессы:

- нагрев формы под заливку;
- тепловые процессы при затвердевании;
- образование усадочных раковин и макропористости;
- образование микропористости;
- гидродинамические процессы при заливке;
- формирование и расчет любых «критериев качества» для прогноза структуры и механических свойств.

Моделирование ведется на базе метода конечных элементов (МКЭ), который позволяет использовать достаточно адекватные физические и геометрические модели (ГМ).

На основании опыта литейного производства можно выделить две задачи, решение которых требует применения программного обеспечения. Первая задача – совершенствование технологии и повышение выхода годных отливок в серийном производстве – требует углубленного учета реальных условий получения отливок с целью выявления причин возникновения дефектов и правильного определения способов их устранения. Вторая задача – разработка литниково-питающей системы (ЛПС) для получения единичных литых заготовок опытных изделий в сжатые сроки без изготовления оснастки. В этом случае ведется расчет ЛПС для стандартных температурно-временных параметров процесса и разрабатывается электронная модель блока отливок для изготовления модельного блока на установках быстрого прототипирования.

В серийном производстве проведение анализа с использованием моделирования в ряде случаев позволило получать качественные отливки и выработать рекомендации по снижению уровня брака по усадочным дефектам. В качестве примера можно привести моделирование технологического процесса рабочей лопатки 4 ступени силовой турбины наземной установки (рис. 2.3).

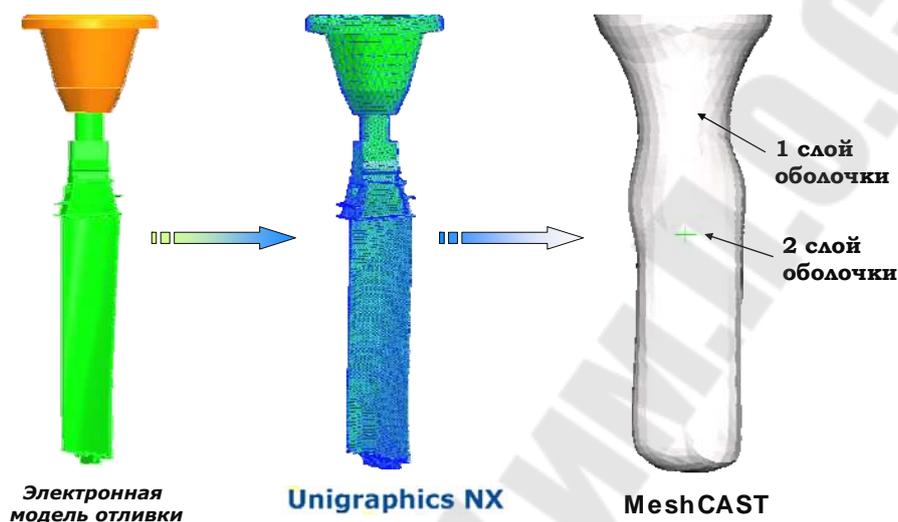


Рис. 2.3. Рабочая лопатка 4-й ступени силовой турбины (серийное производство)

Процесс моделирования состоит из нескольких этапов:

1. На первом этапе в CAD-системе (мы используем Unigraphics) строится трехмерная электронная модель отливки, на которую наносятся припуски и создается литниково-питающая система.

2. Построение сетки для электронной модели отливки.

3. Генерация конечно-элементной сетки в программе Altair HyperMesh 3D.

4. Подготовка геометрической модели в генераторе сетки MeshCAST (модуль системы ProCAST): это построение оболочки для модели и создание 3D конечно-элементной сетки модели.

5. Файл, содержащий 3D конечно-элементную сетку модели, из генератора сетки MeshCAST импортируется в САМ ЛП «Полигон».

6. В системе «Полигон» уже моделируется нагрев формы, тепловые процессы при затвердевании и образование макро- и микропористости в отливке. Все файлы с результатами расчетов можно визуально наблюдать в каждый расчетный момент времени.

Проведенный расчет показывает, что мы получим отливку с минимальным количеством усадочных дефектов. Как видно, замок, и

перо лопатки получились чистыми, только в нижней части пера лопатки имеется пористость до 0,1 % (рис. 2.4).

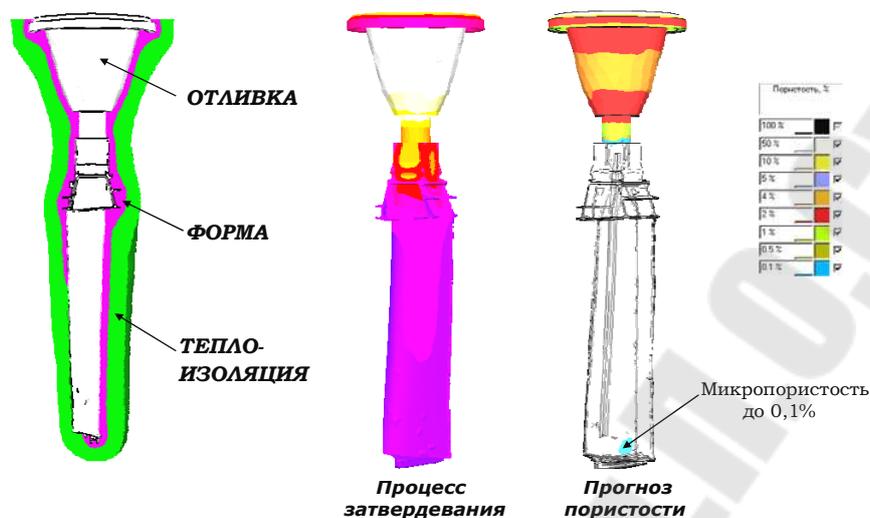


Рис. 2.4. Результаты моделирования в цветокодированном виде для отливки «Рабочая лопатка 4-й ступени силовой турбины» (серийное производство)

В опытном производстве проведение анализа с использованием моделирования позволило получать первые экземпляры опытных отливок в короткие сроки.

Для примера на рис. 2.5 показана керамическая огнеупорная оболочка, изготовленная по фотополимерной модели без использования оснастки.



Рис. 2.5. Результаты моделирования в цветокодированном виде для отливки «Рабочая лопатка 4-й ступени силовой турбины» (серийное производство)

Анализ литейного процесса для детали «Сегмент» был проведен с целью выработки рекомендации для снижения дефектов усадочного происхождения (рис. 2.6).

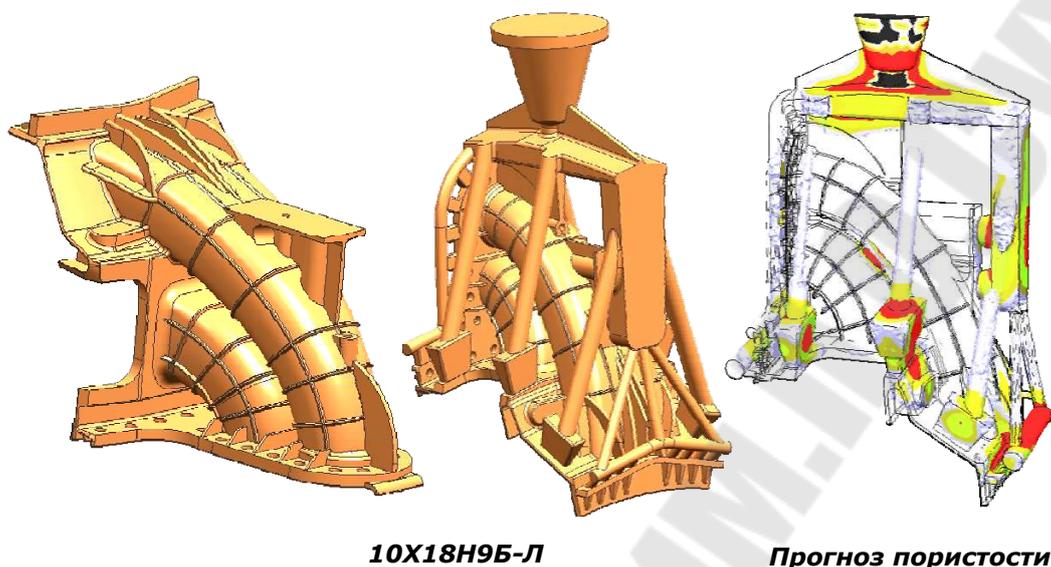


Рис. 2.6. Результаты моделирования в цветокодированном виде для отливки «Сегмент» (опытное производство)

Деталь изготавливают литьем по выплавляемым моделям из нержавеющей стали 10X18H9Б-Л. Конструкция ЛПС, предложенная специалистами литейного цеха на основании практического опыта, не обеспечила получение плотной и качественной отливки. Кроме того, литниково-питающая система при затвердевании деформировала отливку.

Вычислительный эксперимент в системе «ProCAST», воспроизводящий условия реального процесса заливки литейного блока в цехе, дал аналогичные результаты: наличие макро- и микропористости в отливке, в некоторых местах непроливы.

Моделирование гидродинамики заполнения формы расплавом показало, что заполнение формы металлом идет не так, как предполагалось. Автор предполагал, что металл будет одновременно заполнять фланцы и центральную часть отливки («трубы»). Выяснилось, что металл, прежде всего, заполняет стенки «труб» и только, после того, как это становится невозможным из-за кристаллизации металла, металл начинает поступать в питатели, ведущие к фланцам (рис. 2.7).



Рис. 2.7. Гидродинамика заполнения формы расплавом для отливки «Сегмент» (опытное производство)

Результаты этого расчета позволили внести изменения не только в ЛПС, но и в конструкцию отливки, улучшающие процесс заполнения формы, сохраняя при этом то полезное, что было в прежней литниковой системе. Новая ЛПС доставляет расплав к нижнему фланцу, что улучшает условия заполнения отливки, обеспечивает подвод металла к тем местам, где наблюдались непроливы.

Усилено питание мест формирования усадочной пористости. Для этого изменена форма прибылей на фланцах и подведено питание к проблемным зонам отливки (рис. 2.8).



Рис. 2.8. Оптимизация технологии по результатам моделирования. Отливка «Сегмент» (опытное производство)

В качестве еще одного примера возьмем рабочее колесо паровой турбины. Материал, из которого изготовлялась отливка, ЧС88. Специалистами в литейном цехе была разработана технология изготовления отливки, которая не дала положительных результатов. Отливка была поражена пористостью (рис. 2.9).

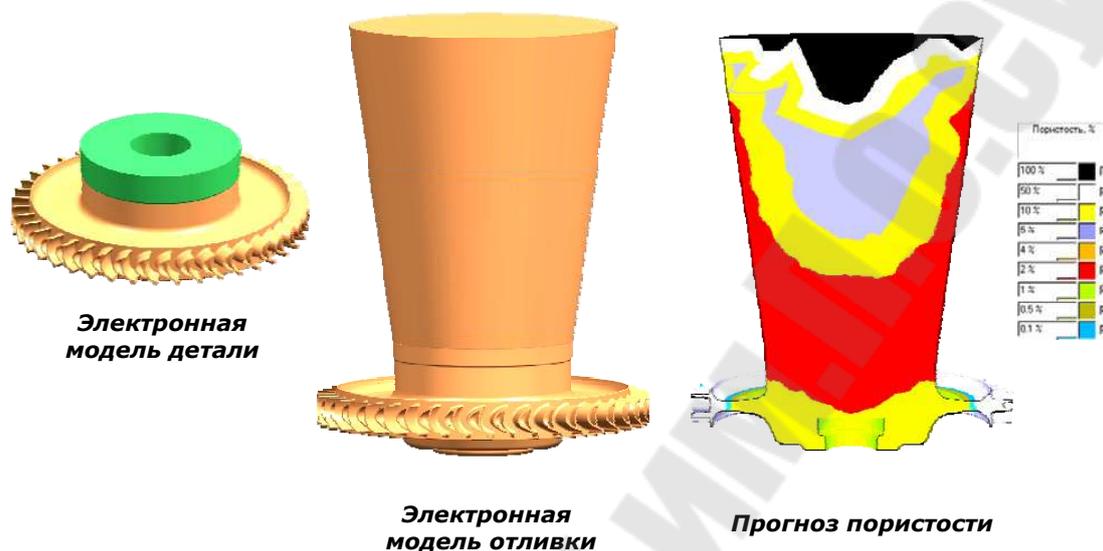


Рис. 2.9. Рабочее колесо паровой турбины (исходный вариант технологии, опытное производство)

Для определения причин образования усадочных дефектов в системе «Полигон» был промоделирован процесс литья с конструкцией литейного блока и технологическими параметрами, используемыми в производстве. Результаты расчетов также показали, что в самой отливке имеется пористость от 1 до 2 %.

Анализ моделирования показал, что из-за широкого интервала кристаллизации в основном вся отливка находится в двухфазном состоянии, включая прибыль. Поэтому прибыль уже не может питать отливку. Чтобы отливка смогла закристаллизоваться раньше прибыли, нужно было прибыль отделить от отливки. Было принято решение сделать прибыль «на ножке». А внутри в полость отливки установить стержень. При такой конструкции ЛПС жидкая фаза скапливается в шарообразной прибыли, что обеспечивает питание отливки. В итоге это обеспечило получение качественной и плотной отливки (рис. 2.10).

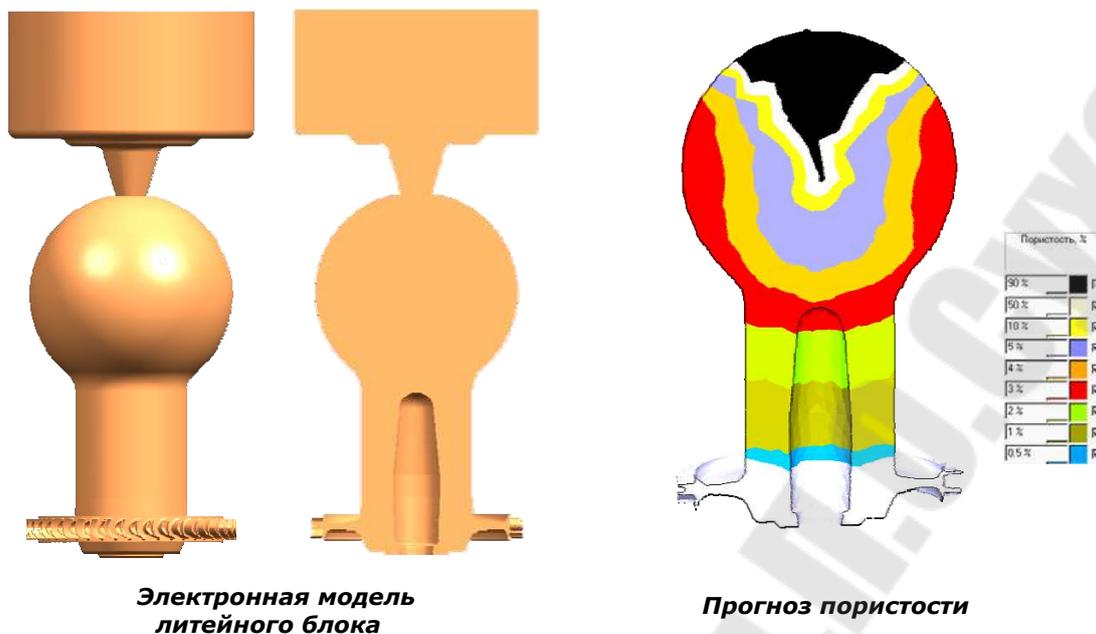


Рис. 2.10. Рабочее колесо паровой турбины (оптимизированный вариант технологии, опытное производство)

Для меньшего расхода металла, основываясь на данной технологии, была разработана конструкция ЛПС на три отливки. Это позволило снизить расход металла примерно на 135 кг. Прогноз пористости, рассчитанный для трехместного литейного блока, показывает, что мы получаем плотную отливку (рис. 2.11).

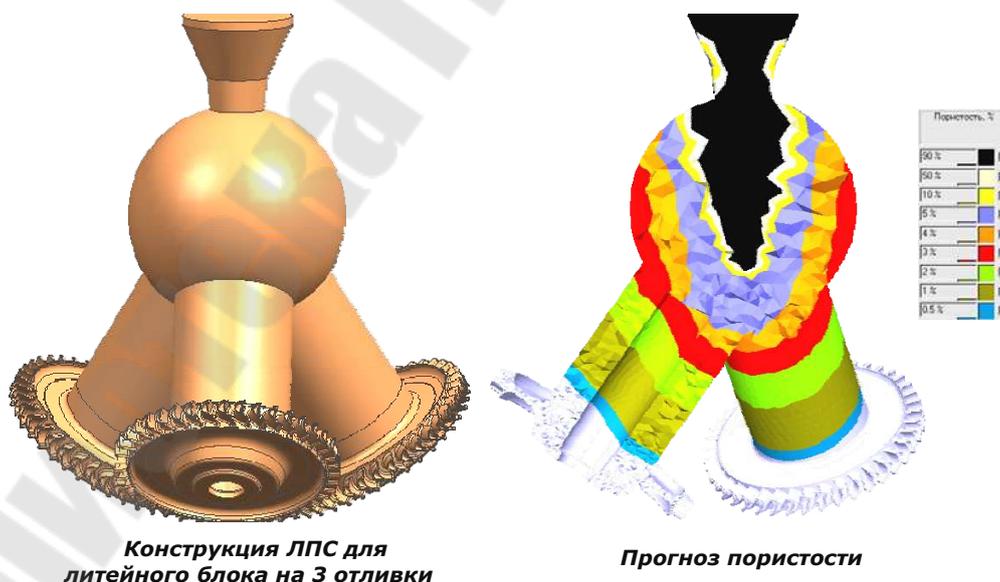


Рис. 2.11. Рабочее колесо паровой турбины. Прогноз пористости, рассчитанный для трехместного литейного блока (оптимизированный вариант технологии, опытное производство)

2.3. Структура отдела САПР литейного производства

В структуру отдела САПР ЛП входят два бюро: конструкторское и технологическое.

Конструкторское бюро выполняет следующие задачи:

- разработка 3D-моделей отливок для участка быстрого прототипирования;

- разработка 3D-моделей деталей по чертежам;

- подготовка моделей для проведения расчетов.

В обязанности технологического бюро входит:

- разработка конструкции ЛПС;

- проведение расчетов и подготовка рекомендаций;

- согласование с литейными цехами технического задания на проведение расчетов;

- проработка чертежа отливки и согласование его с цехами;

- отработка технологического процесса (участие в сборке модельного блока – эталона, присутствие на заливке первых блоков, оценка качества полученной отливки);

- авторский надзор за технологическим процессом при изготовлении первого комплекта;

- ведение базы данных по теплофизическим характеристикам материалов, литейному оборудованию.

Взаимодействие отдела САПР ЛП с литейным производством регламентируется производственной инструкцией, определяющей обязанности и порядок запуска в производство деталей опытных изделий (рис. 2.12).

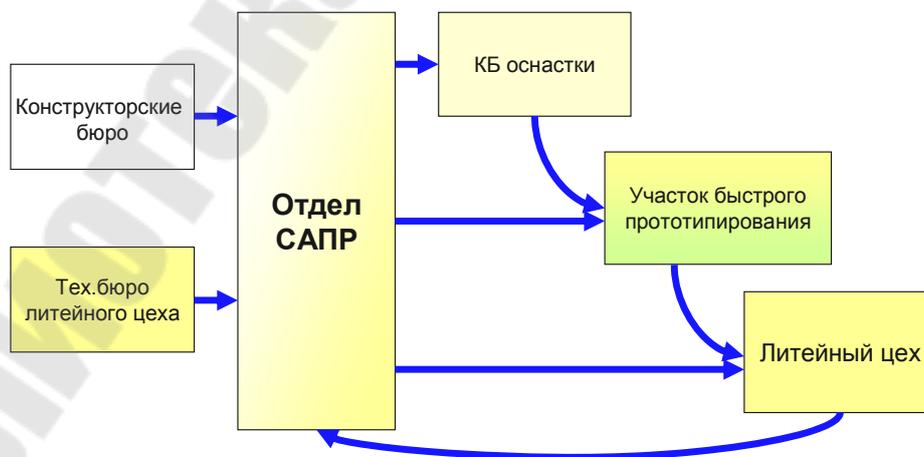


Рис. 2.12. Взаимодействие отдела САПР ЛП с литейным производством

1. В отдел САПР ЛП поступает задание на моделирование ЛПС отливки.

2. По электронной модели детали, которая берется в конструкторском бюро, разрабатывается электронная модель отливки.

3. Технологическим бюро литейного цеха в отдел САПР ЛП выдаются технические указания на отливку для моделирования (припуски на механическую обработку, свой вариант ЛПС) и условия заливки (температура металла, температура формы, марка сплава, способ заливки, условия охлаждения формы и т. п.).

4. На основании технического задания в отделе САПР ЛП разрабатывается электронная модель ЛПС и конструкция литейного блока. Проводится моделирование процесса заливки металла в литейный блок и его кристаллизации в системах «Полигон» и ProCAST. Делается прогноз возможных усадочных дефектов в отливках и, если необходимо, меняется конструкция литейного блока и температурно-временные параметры процесса.

5. После получения положительного результата электронная модель литейного блока передается на участок быстрого прототипирования, где на установке SLA-7000 или ThermoJet выращивается модель литейного блока. Модель передается в литейный цех для изготовления керамической оболочки или используется в качестве мастер-модели для изготовления временной прессформы.

6. После изготовления керамической формы и заливки, полученная деталь подвергается необходимым контрольным операциям. По результатам контроля отливки делается заключение о качестве отливки, принимается решение о пригодности опробованной технологии литья и выдается заказ на изготовление постоянной оснастки.

Проработка ЛПС с помощью систем «Полигон» и ProCAST в несколько раз сокращает время и затраты на отработку технологии литых заготовок для деталей опытных изделий, поскольку позволяет резко сократить число опытных плавков.

3. ОБРАБОТКА ДАННЫХ В СИСТЕМЕ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

3.1. Стандарты обмена данными между системами

Прикладные программы, например, программы генерации сетки для анализа по методу конечных элементов или траектории движения инструмента станков с ЧПУ, требуют на входе техническое описание продукта. Данные технических требований делятся на два типа. Первый тип данных – это данные чертежа; они включают векторное описание линий (сплошных, пунктирных, осевых, размерных и выносных) и пояснительных данных (комментариев, символов и значений размеров), имеющих на чертеже.

Ко второму типу данных технических требований относится представление твердотельной модели и некоторые пояснительные данные. Поэтому данные технических требований обычно импортируются из САД-системы – либо из системы автоматизированной разработки чертежей, либо из системы твердотельного моделирования. Однако все САД-системы хранят результаты проектирования, т. е. данные технических требований, в своих собственных структурах данных, формат которых зависит от конкретной системы. Они могут не соответствовать входному формату используемой прикладной программы. Таким образом, когда две или более САД/САМ/САЕ-системы объединяются и связываются в единое приложение для совместного использования данных, часто возникает проблема обмена данными. Фактически всегда имеется потребность связать воедино несколько систем либо внутри одной организации, либо внешне, как в случае со смежниками или поставщиками компонентов.

Для решения этой коммуникационной проблемы необходимо преобразовывать данные технических требований одной системы в форму, понятную для других систем, и наоборот. Чтобы облегчить преобразование и не разрабатывать программы-конверторы для всех возможных пар САПР, было предложено несколько стандартных форматов для хранения данных технических требований. Далее мы вкратце рассмотрим типичные стандартные форматы.

3.2. Методы обмена данными технических требований

Различные САД/САМ/САЕ-системы хранят данные технических требований в структурах разного вида, поэтому для переноса данных

необходимо преобразовать данные технических требований одной системы в формат другой системы. Еще один конвертор необходим для переноса данных между двумя системами в противоположном направлении. Следовательно, для каждой пары систем необходимо иметь два конвертора. Двухнаправленные стрелки для каждой пары систем (рис. 3.1, а) предполагают наличие двух конверторов. Эти конверторы для каждой конкретной пары систем называются *прямыми конверторами* (direct translators). Если у нас есть n различных систем, нам необходимо разработать $n(n - 1)$ конверторов, поскольку количество пар систем равно $n(n - 1)/2$. Например, для обмена данными между 10 системами придется разработать 90 конверторов. Таким образом, метод прямого конвертирования непрактичен, т. к. требует разработки слишком большого количества конверторов при необходимости работать с множеством систем. Более того, добавление одной системы к n уже имеющихся, потребует написания $2n$ дополнительных конверторов.

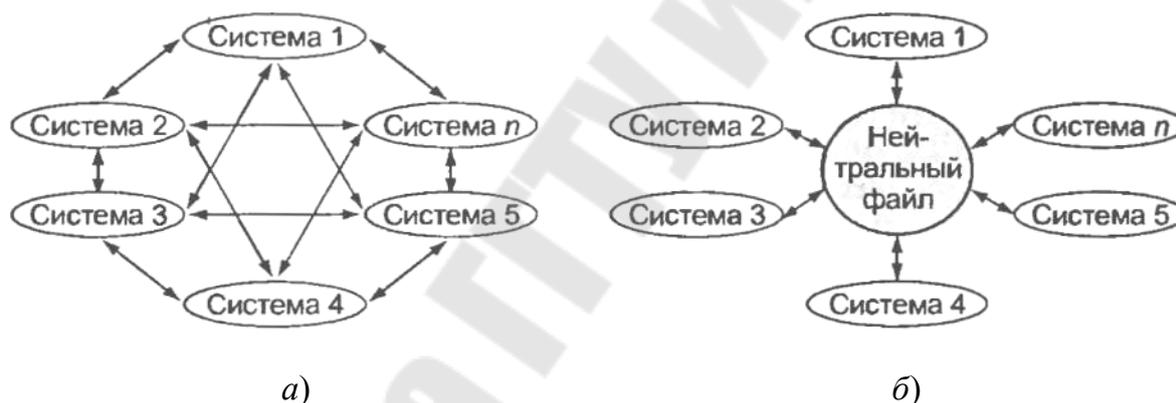


Рис. 3.1. Два метода обмена данными между двумя различными системами

Однако обмен данными можно обеспечить, введя нейтральную структуру базы данных, называемую *нейтральным файлом* (neutral file), которая была бы независима от существующих САПР. Эта структура будет действовать как промежуточная точка коммуникации между различными структурами баз данных САПР (рис. 3.1, б). Таким образом, в каждой системе будет своя пара конверторов для экспорта и импорта данных в этот нейтральный формат. Конвертор, преобразующий данные из собственного формата данной системы в нейтральный формат, называется *препроцессором* (pre-processor), а конвертор, выполняющий обратное преобразование – *постпроцессором* (post-processor). Соответственно, в этом случае для обмена данными между n системами

потребуется $2n$ конверторов, и лишь два дополнительных конвертора необходимо будет добавить при введении новой системы. Иными словами, этот косвенный метод свободен от присущего прямому методу недостатка, когда требовалось писать все возрастающее количество программ. Это основная причина, по которой косвенный метод принят в качестве главного метода обмена данными между различными системами, несмотря на то, что в сравнении с прямым методом у него имеются некоторые недостатки. В частности, прямые конверторы работают быстрее косвенных, и создаваемые ими файлы данных обычно имеют меньший размер, чем нейтральные файлы, генерируемые косвенными конверторами. Файл данных в собственном формате конкретной системы обычно также оказывается меньше нейтрального файла из-за обобщенной природы последнего. Когда мы переносим данные технических требований через нейтральный файл, некоторая информация, как правило, теряется, особенно информация о топологическом дереве и ограничениях в системах параметрического моделирования.

3.3. Форматы файлов обмена данными в САПР

В этом подразделе будут рассмотрены три типичных формата нейтрального файла: IGES (Initial Graphics Exchange Specification – первоначальная спецификация обмена графическими данными), DXF (Drawing interchange Format – формат обмена чертежами) и STEP (STandard for Exchange of Product model data – стандарт обмена данными о модели продукта). В настоящее время IGES является самым популярным форматом нейтрального файла, а формат DXF используется главным образом для обмена данными чертежей. STEP – это стандартный формат данных, используемый для хранения полной информации обо всем жизненном цикле продукта, включая проектирование, анализ, производство, контроль качества, испытания и обслуживание помимо обычных данных технических требований. В настоящее время САД-системы, поддерживавшие формат IGES, ориентированы на переход к формату STEP.

3.3.1. Формат IGES

В 1979 г. перед техническим комитетом, который состоял из компании Boeing, компании General Electric и Национального бюро стандартов США (National Bureau of Standards, теперь Национальный институт стандартов и технологии), была поставлена задача разработать метод обмена данными в рамках программы интегрированного

автоматизированного производства (ICAM) для ВВС США. Результатом этих усилий явилось описание формата IGES версии 1.0, опубликованное в январе 1980 г. В 1981 г. оно было принято Американским Национальным институтом стандартов (ANSI) в качестве стандарта.

IGES был первым стандартным форматом обмена данными, разработанным для нужд передачи данных технических требований между различными САПР. Ранние версии IGES были неявным образом ориентированы на CAD/CAM-системы 1970-х и начала 1980-х гг., т. е. главным образом на обмен чертежами. В более поздних версиях спектр типов данных, подлежащих обмену, был расширен. Например, версия 2.0 поддерживала обмен данными анализа по методу конечных элементов и данными печатных плат, в версии 3.0 были расширены возможности пользовательских макрокоманд, играющих важную роль при обмене стандартными библиотеками деталей, в версии 4.0 была введена поддержка дерева CSG, в версии 5.0 появилась обработка данных структуры B-Rep.

IGES-файл состоит из шести разделов, которые должны идти в следующем порядке: Flag (Флаг, необязательный раздел), Start (Начало), Global (Глобальные данные), Directory Entry, или DE (Запись в каталоге), Parameter Data, или PD (Параметрические данные) и Terminate (Конец). Пять обязательных разделов идентифицируются буквами S, G, D, P и T. *Запись* (record) – это строка из 80 символов.

Раздел Flag используется только в сжатом ASCII-формате и бинарном формате. Данные в IGES-файле могут быть представлены в двух форматах: ASCII и бинарном. Формат ASCII имеет две разновидности: фиксированную длину строки 80 символов и сжатую форму. Сжатая форма – это не что иное, как ASCII-файл, сжатый путем устранения пробелов между записями. Бинарный формат файла являет собой бинарное представление данных в виде потока битов в формате с фиксированной длиной записи. Чтобы идентифицировать формат файла как сжатый ASCII, в столбец 73 раздела Flag записывается символ C. Раздел Flag состоит из одной записи (строки) и предшествует разделу Start. В бинарном формате первый байт (8 бит) раздела Flag содержит ASCII-символ B, служащий идентификатором типа файла. В разделе Start дается описание файла в форме, воспринимаемой человеком. В нем указывается система, являющаяся источником данных, препроцессор и описываемый продукт.

Раздел Global содержит информацию о препроцессоре, а также информацию, необходимую постпроцессору для интерпретации файла. В частности, в этом разделе имеются следующие элементы:

- символы, используемые в качестве разделителей между отдельными полями и записями (для этого используются соответственно запятая и точка с запятой);
- имя самого IGES-файла;
- количество значащих цифр в представлении целых чисел и чисел с плавающей точкой в системе-источнике;
- дата и время создания файла;
- масштаб пространства модели;
- единицы измерения модели;
- минимальная разрешающая способность и максимальные значения координат;
- имя создателя файла и название организации.

Раздел Directory Entry содержит список всех элементов и некоторых их атрибутов. В IGES-файле все данные технических требований представлены в виде списка predetermined элементов: геометрических (линии, кривые плоскости, поверхности) и пояснительных (комментарии и значения размеров). Каждому элементу присваивается определенный номер типа.

Раздел Parameter Data содержит фактические данные, описывающие каждый из элементов, перечисленных в разделе Directory Entry. Например, элемент, представляющий собой прямую линию, определяется шестью координатами двух ее конечных точек. Хотя каждый элемент всегда имеет две строки в разделе DE, количество записей, требуемое для каждого элемента в разделе PD, различно для разных элементов и зависит от количества необходимых данных. Параметрические данные указываются в свободном формате в столбцах 1–64. Разделитель полей, определенный в разделе Global, используется для разделения параметров, а определенный там же разделитель записей – для обозначения конца списка параметров. Обычно в качестве разделителя полей используется запятая, а в качестве разделителя записей – точка с запятой. Столбцы 66–72 во всех записях раздела PD содержат обратный указатель на соответствующий элемент в разделе DE. Обратите внимание, что этот указатель представляет собой порядковый номер первой из двух строк, отведенных под каждый элемент в разделе DE. В столбцах 74–80 хранится порядковый номер записи, которой они принадлежат. Раздел Terminate содержит единственную запись, в которой в целях контроля хранится количество записей в каждом из четырех предшествующих разделов.

При использовании препроцессоров и постпроцессоров с нейтральным форматом IGES на практике возникают следующие про-

блемы. Во-первых, внутренний способ представления элемента в системе может отличаться от того, как этот элемент представляется в IGES. Например, дуга окружности в какой-то системе может быть определена через центр, радиус и начальный и конечный углы, но в IGES она определяется через центр, начальную точку и конечную точку. Таким образом, специализированный IGES-конвертор должен выполнить преобразование с использованием параметрического уравнения дуги. Такое преобразование должно выполняться дважды (при прямой и обратной конвертации), и каждый раз значения параметров дуги искажаются из-за ошибок усечения и округления.

Вторая проблема более серьезная: она возникает, когда элемент не поддерживается явно, и поэтому его необходимо преобразовать в ближайший по форме доступный элемент. Эта проблема часто имеет место при обмене данными между двумя системами через IGES-файл, если конверторы этих систем поддерживают разные версии IGES. Типичный пример – потеря символьной информации в случае, когда одна из двух систем использует более старую версию IGES, не поддерживающую макросы.

3.3.2. Формат DXF

Формат DXF (Drawing interchange Format – формат обмена чертежами) изначально разрабатывался для того, чтобы предоставить пользователям гибкость в управлении данными и преобразовании чертежей программы AutoCAD в форматы файлов, которые могли читаться и использоваться другими САПР. Из-за популярности AutoCAD формат DXF стал фактическим стандартом обмена файлами САД-чертежей почти для всех САПР. На самом деле почти в каждой из появляющихся новых САПР имеется транслятор в формат DXF и обратно.

DXF-файл – это текстовый ASCII-файл, состоящий из пяти разделов: Header (Заголовок), Table (Таблица), Block (Блок), Entity (Элемент) и Terminate (Конец). В разделе Header описывается среда AutoCAD, в которой был создан DXF-файл. В разделе Table содержится информация о типах линий, слоях, стилях текста и видах, которые могут быть определены на чертеже. В разделе Block содержится список графических элементов, определенных как группа. Конкретные данные по каждому элементу хранятся в соответствующем разделе Entity, который следует сразу за разделом Block. Раздел Entity – это главный раздел DXF-файла, в котором описываются все элементы, присутствующие на чертеже.

Аналогично тому, как это происходило с IGES-файлами, с появлением новых версий AutoCAD список возможных элементов DXF-файлов расширялся. DXF-файл, созданный более поздней версией AutoCAD, не может быть прочитан другими системами, использующими более старые версии формата DXF.

3.3.3. Формат STEP

Форматы IGES и DXF были разработаны для обмена данными технических требований, а не данными о продукте. Под данными о продукте мы понимаем данные, относящиеся ко всему жизненному циклу продукта (например, проектирование, производство, контроль качества, испытания и поддержка). Хотя спецификации IGES и DXF были расширены с целью включения некоторых из этих данных, информации, содержащейся в этих файлах, по существу недостаточно для описания всего жизненного цикла продукта. Вследствие этого в США в 1983 г. началась разработка нового стандарта под названием PDES (Product Data Exchange Specification – спецификация для обмена данными о продуктах). Основной упор в PDES делался не на обмен данными технических требований, а на то, чтобы исключить человеческое присутствие из обмена данными о продукте. Иначе говоря, целью PDES было устранить потребность в инженерных чертежах и других бумажных документах при обмене информацией о различных фазах жизненного цикла продукта между сходными или различающимися САПР. Между тем в июле 1984 г. в Международной организации по стандартизации (ISO) были образованы технический комитет TC 184 (Системы промышленной автоматизации) и его подкомитет SC4 (Внешнее представление данных о модели продукта) для установления единого международного стандарта обмена данными о модели продукта – STEP (STandard for Exchange of Product model data). Цели PDES и STEP были идентичны, поэтому в июне 1985 г. Управляющий комитет IGES решил, что интересы США в программе STEP должен представлять стандарт PDES. В результате значение акронима PDES поменяли на «обмен данными о продукте с использованием STEP» (Product Data Exchange using STEP), чтобы подчеркнуть идентичность целей PDES и STEP.

В основе разработки STEP лежат следующие принципы.

1. Стандарт STEP должен ориентироваться на данные о продукте, которые включают информацию обо всем жизненном цикле продукта: проектировании, производстве, контроле качества, испытании

и поддержке. Таким образом, в качестве данных должна рассматриваться информация о допусках, технологических особенностях формы, конечноэлементная модель, модель для кинематического анализа и т. д., а также данные технических требований, относящиеся главным образом к форме продукта.

2. В структурах данных STEP информация, относящаяся к приложению, должна храниться в модуле уровня приложения, отдельно от общей информации о форме. Благодаря такому подходу структура данных сможет поддерживать широкий спектр приложений, избегая при этом избыточности в общей структуре данных.

3. Для определения структуры данных должен использоваться формальный язык. Спецификации IGES и DXF описывают формат физического файла, в котором хранятся все геометрические и прочие данные. В STEP данные описываются на языке EXPRESS, а затем результат преобразовывается в физический файл. Таким образом можно избежать неоднозначностей при интерпретации данных о продукте, извлеченных из файла.

STEP разрабатывается рядом комитетов и рабочих групп, занимающихся разными *частями* стандарта. Эти части группируются по методам описания, интегрированным информационным ресурсам, прикладным протоколам, методам реализации и методологией согласования. Статус каждой части показан рядом с ее номером. Статус обозначается буквами от «O» (предварительная стадия ISO) до «I» (международный стандарт – высшая стадия разработки и принятия стандартов). Части, обозначенные буквами «E», «F» (проект международного стандарта) и «I», считаются находящимися на достаточно высоком уровне для того, чтобы позволить производителям программных продуктов приступить к их реализации.

Группа методов описания образует фундамент STEP. Она включает часть 1 «Обзор», которая содержит также определения, являющиеся в STEP универсальными. Принадлежащая той же группе часть 11 «Справочное руководство по языку EXPRESS» описывает язык моделирования данных, который используется в STEP. Части, относящиеся к группе методов описания, имеют номера от 1 до 9.

На следующем уровне находится группа интегрированных информационных ресурсов – части, содержащие фактическое описание моделей данных STEP. Эти модели данных являются «кирпичиками» STEP. Интегрированные информационные ресурсы включают обобщенные ресурсы, прикладные ресурсы и конструкции, интерпретируемые приложением (рис. 3.2).

Интегрированные прикладные ресурсы	
I 101 Черчение X 102 Корабельные структуры X 103 BE – соединение C 104 Конечноэлементный анализ	I 105 Кинематика W 106 Построение базовой модели A 107 Разработка аналитической модели
Интегрированные обобщенные ресурсы	
I 41 Основное описание и спецификация продукта I 42 Геометрическое и топологическое представление продукта I 43 Уточняющее представление I 44 Структурная конфигурация продукта	F 45 Материалы I 46 Визуальное представление F 47 Допуски X 48 Характеристики формы F 49 Структура и свойства процесса
Конструкции, интерпретируемые приложением	
C 501 Реберный каркас C 502 Оболочечный каркас C 503 Геометрические ограничения двумерного каркаса C 504 Надписи на чертеже C 505 Структура чертежа C 506 Элементы чертежа C 507 Геометрические ограничения поверхности C 508 Незамкнутые поверхности C 509 Трубчатые поверхности C 510 Геометрические ограничения каркаса C 511 Топологические ограничения каркаса	C 512 Ячеистое граничное представление C 513 Элементарное граничное представление C 514 Расширенное граничное представление C 515 Конструктивная твердотельная геометрия C 516 Контекст проектирования механических объектов C 517 Геометрическое представление механических объектов C 518 Объемное представление механических объектов C 519 Геометрические допуски C 520 Ассоциированные элементы чертежа

Рис. 3.2. Архитектура STEP. Интегрированные информационные ресурсы

Интегрированные обобщенные ресурсы – это элементы, которые используются по необходимости прикладными протоколами. Номера частей, относящихся к обобщенным ресурсам, начинаются с 40 и используются всей гаммой прикладных протоколов STEP. Интегрированные прикладные ресурсы содержат элементы, имеющие несколько больший объем контекста, чем обобщенные элементы. Номера частей, относящихся к прикладным ресурсам, начинаются со 100. Части с номерами от 500 – это конструкции, интерпретируемые приложением. Они представляют собой многократно используемые группы информационных ресурсов, облегчающие представление одной и той же семантики в различных прикладных протоколах.

На верхнем уровне иерархии STEP находятся более сложные модели данных, используемые для описания конкретных данных о про-

дукте. Эти части называются прикладными протоколами и описывают не только то, какие данные должны использоваться при описании продукта, но и то, как эти данные должны использоваться в модели. Прикладные протоколы используют интегрированные информационные ресурсы в четко очерченных сочетаниях и конфигурациях для представления определенной модели данных или некоторой фазы жизненного цикла продукта. Прикладные протоколы нумеруются, начиная с 200. В настоящее время используются такие прикладные протоколы, как «Явное черчение» (201) и «Проектирование с управлением конфигурацией» (203).

Группа методов реализации STEP, части в которой нумеруются с 20, описывает соответствие между формальными спецификациями STEP и представлением, используемым для реализации STEP. Группа методологии проверки соответствия, части в которой нумеруются с 30, предоставляет информацию о методах проверки соответствия программных продуктов стандарту STEP, дает указания по созданию абстрактных испытательных пакетов и описывает задачи испытательных лабораторий. Часть 31, описывающая методологию выполнения проверки соответствия, принята в качестве международного стандарта. Стандарты STEP уникальны в том отношении, что они делают упор на испытания и содержат в себе описания методов испытаний.

Группа частей с номерами от 300 (абстрактные испытательные пакеты), состоит из данных и критериев, используемых для проверки соответствия программного продукта, реализующего стандарт STEP, его прикладному протоколу. Номера, присваиваемые абстрактным испытательным пакетам, превышают номера прикладных протоколов ровно на 100. Таким образом, абстрактный испытательный пакет с номером 303 относится к прикладному протоколу 203 (Более подробную информацию о STEP можно получить на сайте <http://www.nist.gov/sc4>).

Сегодня STEP привлекает к себе повышенное внимание, т. к. ожидается, что он войдет в систему стандартов технологий CALS (Computer-aided Acquisition and Logistics Support – Непрерывные поставки и информационная поддержка жизненного цикла продукции) как стандарт обмена данными о продуктах. Цель инициативы CALS – компьютеризация процесса формирования требований, заказа, эксплуатации, поддержки и обслуживания систем. Основное внимание эта инициатива уделяет заданию форматов, которые будут использоваться для хранения и обмена компьютерными данными. Хотя CALS создавалась для военных целей, она становится промышленным стандартом хранения и обмена компьютерными данными в организации.

4. ОСНОВНЫЕ КОНЦЕПЦИИ ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ. СИСТЕМЫ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКИ ЧЕРТЕЖЕЙ

4.1. Основные концепции графического программирования

Интерактивное манипулирование формами составляет значительную часть работы с системами CAD/CAM/CAE. Следовательно, важнейшей составляющей таких систем является программное обеспечение, создающее графическое изображение на экране монитора. Поэтому нам придется изучить терминологию и основные понятия графического программирования.

Термин *программирование на компьютере* (computer programming) раньше означал написание «сочинения» на языке компьютерных команд в соответствии с predetermined правилами грамматики. В ответ на вводимые числа выполняемое «сочинение» порождало ожидаемые числа и символы на терминале или в файле данных. В наши дни на входе и выходе «сочинений» все чаще находится графическая информация. Такое программирование называется *графическим* (graphics programming), а область его применения – *компьютерной графикой* (computer graphics).

Помимо основного программного обеспечения, необходимого для обычного программирования (операционная система, редактор и компилятор), графическое программирование требует наличия специальных графических программ. Графические программы делятся на два класса: драйверы устройств и графические библиотеки.

Драйвер устройства может рассматриваться как набор аппаратно-зависимых кодов, непосредственно управляющих процессором графического устройства. Драйверы обязательно являются аппаратно-зависимыми, т. е. жестко привязанными к конкретным графическим процессорам. Примерно то же самое можно сказать об ассемблере, конкретный вид которого может выполняться только на процессорах одной и той же модели. То же происходит, если графическая программа использует драйвер устройства непосредственно (рис. 4.1). Такую графическую программу при переходе на другое графическое устройство придется переписывать с использованием новых команд драйвера. Более того, команды драйвера устройства весьма примитивны, поэтому такая программа была бы очень длинной, если бы она

должна была решать какую-либо сложную задачу. К тому же программа эта получилась бы плохо читаемой. Программисты предпочитают писать программы на языках высокого уровня. Графическое программирование не могло стать исключением, особенно если представить все неудобства, связанные с использованием команд драйвера низкого уровня. Поэтому с графическими устройствами стали поставляться библиотеки, получившие название *графических* (graphics libraries). Графическая библиотека, как, например, и математическая, представляет собой набор подпрограмм, предназначенных для решения определенных задач. Конкретная подпрограмма может изображать на экране прямую, круг или иной объект. Графическая библиотека основывается на командах драйвера устройства (рис. 4.2). Каждая подпрограмма создается с использованием поддерживаемого набора команд драйвера. Например, подпрограмма, изображающая круг, может быть составлена из отдельных команд драйвера, рисующих на экране точки или короткие отрезки.



Рис. 4.1. Непосредственное использование драйвера устройства



Рис. 4.2. Использование графической библиотеки

Подпрограммы графической библиотеки могут использоваться точно так же, как подпрограммы математической. Нужная подпрограмма вызывается из основной программы аналогично тому, как вызываются функции синуса и косинуса, когда программисту требуется вычислить их значения. Одна из проблем использования подпрограмм графической библиотеки связана с тем, что их названия и способы вызова (входные и выходные аргументы) у каждой библиотеки свои. Это не создавало бы трудностей, если бы одна графическая библиотека могла работать со всеми существующими устройствами, что теоретически было бы возможным, если бы все существующие драйверы устройств поддерживали ее. Однако по некоторым причинам производители программного обеспечения не хотят или не могут создать

графическую библиотеку, которая могла бы работать со всеми драйверами, а потому у каждой библиотеки имеется свой круг поддерживаемых драйверов. Следовательно, такая библиотека может работать лишь с ограниченным набором графических устройств, а графические программы, рассчитанные на работу с множеством устройств, приходится переписывать с использованием нескольких библиотек.

Чтобы обойти эту проблему, разработчики графических библиотек могли бы использовать одинаковые наборы подпрограмм с одинаковыми названиями, аргументами и возможностями. Тогда графические программы не требовали бы изменения на уровне исходного кода даже при изменении графических устройств. Одним из примеров такого подхода является графическая система CORE, предложенная в 1977 г. группой SIGGRAPH (Special Interest Group on Computer Graphics) Ассоциации вычислительной техники (Association for Computing Machinery – ACM). Однако графическая система CORE не предоставляет достаточного набора команд для использования всех возможностей растровых устройств, потому что эти устройства не были широко распространены во времена разработки системы. Примерно в то же время Международная организация по стандартизации (International Standards Organization – ISO) разработала систему графического ядра (Graphics Kernel System – GKS). GKS считается стандартом в двумерной графике, а позже эта система была расширена до GKS-3D с поддержкой трехмерной графики.

И у CORE, и у GKS есть определенные недостатки в отношении динамического отображения и гибкого взаимодействия с пользователем. Поэтому организация ISO предложила еще один стандарт, получивший название *Иерархическая система программирования интерактивной графики* (Programmer's Hierarchical Interactive Graphics System – PHIGS). Этот стандарт стал стандартом де-факто для большинства графических рабочих станций. Затем он был расширен и получил название «расширение PHIGS для X» (PHIGS extension to X – PEX), потому что в него были включены функции работы с окнами системы X window. Графические программы, написанные на PEX, в сетевой среде могут использоваться независимо от типа рабочей станции. Это достоинство было унаследовано от системы X window, о которой будет коротко рассказано в конце главы. Коммерческая графическая библиотека OpenGL развивалась независимо от организаций по стандартизации, однако с течением времени она набирает все большую популярность благодаря гибкости в управлении рабочими станциями

и персональными компьютерами в сетевой среде. OpenGL – расширение графической библиотеки GL (фирменной графической библиотеки для компьютеров Silicon Graphics). Благодаря популярности компьютеров Silicon Graphics в областях, связанных с компьютерной графикой, библиотека OpenGL постепенно приобретает статус графического стандарта де факто.

4.1.1. Системы координат

Для вывода изображения объекта на экран графического устройства необходимо решить две основные задачи:

- указать положение всех точек объекта в пространстве;
- определить положение их образов на мониторе.

Для задания положения точек в пространстве и на мониторе используются системы координат. Важно понимать, как связаны между собой различные системы координат. Особенно это важно для проектирования трехмерного объекта на плоский экран. Проекция на экране строится по тем же законам, что и проекция реального объекта на сетчатке человеческого глаза.

Первой среди систем координат мы рассмотрим *систему координат устройства* (device coordinate system), которая определяет положение точки на экране. Эта система состоит из горизонтальной оси u и вертикальной оси v (рис. 4.3). Обратите внимание, что начало отсчета может выбираться произвольно. Осей u и v достаточно для задания положения любой точки экрана, поэтому третья ось, перпендикулярная первым двум, не определяется. Положение любой точки задается двумя целыми числами u и v , равными числу пикселей между началом координат и точкой по осям u и v . Однако одна и та же точка может задаваться разными парами u и v в зависимости от положения начала координат, направления осей и масштаба. Эти параметры для разных графических устройств устанавливаются достаточно произвольно (рис. 4.3). Поэтому аппаратные координаты в графической программе могут потребовать изменения при смене графического устройства.

Виртуальная система координат устройства (virtual device coordinate system) позволяет избежать описанной выше проблемы. Виртуальная система координат устройства фиксирует точку отсчета, направление и масштаб осей для всех рабочих станций. Слово «виртуальный» означает, что данная система отсчета существует только в воображении программиста. Обычно начало отсчета располагается

в левом нижнем углу экрана, ось u откладывается вправо, а ось v – вверх. Обе координаты могут изменяться в диапазоне от нуля до единицы. Точка, положение которой задается в виртуальной системе координат, на любом экране будет попадать в одно и то же место. Это дает программисту возможность единообразно определять формы, не заботясь о конкретных системах координат устройств. Графическая программа передает виртуальные координаты подпрограмме драйвера устройства, которая преобразует их в координаты конкретного устройства.

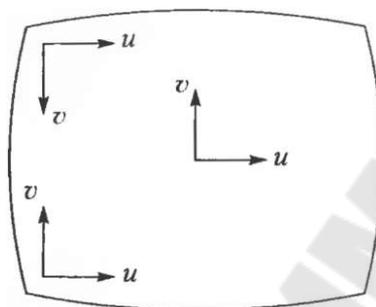


Рис. 4.3. Системы координат устройства

Виртуальная и обычная системы координат устройства позволяют задавать положение точки на плоском экране. Займемся теперь системами координат для работы с трехмерным пространством. Основных трехмерных систем координат всего три: *внешняя система координат* (world coordinate system), *система координат модели* (model coordinate system) и *система координат наблюдателя* (viewing coordinate system).

Внешняя, или мировая система координат (world coordinate system), – это опорная система, используемая для описания интересующего нас мира. Внешней она является по отношению к объектам этого мира. Например, такая система может использоваться для описания расположения и ориентации парт, стульев и доски, если интересующий нас мир представляет собой класс.

Следующим шагом является описание формы каждого объекта мира. Форма объекта определяется координатами всех или некоторых характеристических точек объекта по отношению к системе координат, связанной с ним, – *системой координат модели* (model coordinate system). Координаты точек объекта, определенные таким образом, не изменяются даже тогда, когда объект перемещается или вращается в пространстве. Они действительно зависят только от формы объекта.

Система координат модели перемещается вместе с тем объектом, к которому она привязана. Поэтому форма каждого объекта определяется в его собственной системе координат модели. Расположение и ориентация любого объекта задаются относительным положением и ориентацией модельной системы координат данного объекта по отношению к внешней системе координат. Относительное расположение и ориентация систем координат определяются матрицей преобразования. Наличие внешней системы координат и модельных систем для всех объектов полностью определяет мир, т. е. расположение и форму всех объектов данного мира. Другими словами, применение матриц преобразования позволяет получить координаты любой точки любого объекта во внешней системе.

Следующий шаг – проектирование трехмерных объектов или их точек на монитор подобно тому, как они проектируются на сетчатку человеческого глаза. В компьютерной графике используется два вида проекций: перспективная и параллельная (рис. 4.4).

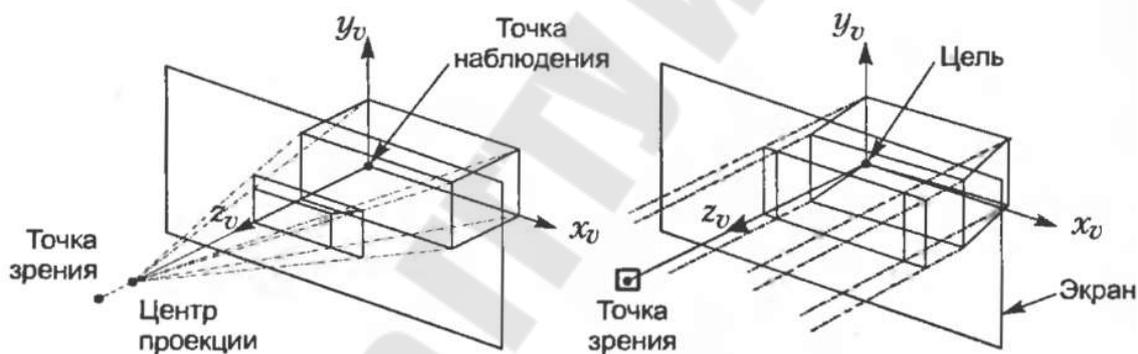


Рис. 4.4. Два вида проекций:
а – перспективная; б – параллельная

Оба вида требуют задания двух точек: точки зрения и точки наблюдения. *Точка зрения* (viewpoint) – это глаз наблюдателя. *Точка наблюдения* (viewsite) – это точка объекта, определяющая направление «луча зрения». Вектор, проведенный от точки зрения к цели, задает направление наблюдения.

В *перспективной проекции* (perspective projection) все точки рассматриваемого объекта соединяются с центром проекции, который обычно лежит на линии, соединяющей точку зрения и цель. Точки пересечения этих линий с экраном образуют проекцию. Экран располагается между точкой зрения и целью. В *параллельной проекции* (parallel projection) линии от всех точек объекта проводятся в направ-

лении наблюдателя параллельно направлению наблюдения, а точки пересечения этих линий с экраном формируют проекцию. Экран, как и в перспективной проекции, располагается перпендикулярно направлению проектирования. Такая проекция называется ортогональной. В противном случае проекция называется косоугольной. В косоугольной проекции ориентация экрана произвольна.

Точки проекции, получаемые любым из описанных методов, легко могут быть рассчитаны, если координаты точек проецируемого объекта даны в системе координат. Например, координаты точек параллельной проекции объекта попросту равняются соответствующим координатам точек объекта.

Системы координат связаны друг с другом матрицами преобразования. Так, положение и ориентация каждой из модельных систем координат по отношению к мировой задаются соответствующими матрицами преобразований. Наблюдательская система координат также может быть определена относительно мировой при помощи матрицы преобразования, если задать положение точек зрения и наблюдения, а также вектор вертикали. Процедура расчета точек проекции с использованием матриц преобразования выглядит следующим образом. Сначала координаты проецируемой точки преобразуются из модельных в мировые при помощи матрицы преобразования, определяющей переход от модели, к которой относится точка, к мировой системе координат. Эта операция (рис. 4.5) называется *преобразованием модели* (model transformation). Затем координаты этой точки преобразуются из мировой системы координат в наблюдательскую. Эта операция называется *преобразованием наблюдения* (viewing transformation).

Все эти преобразования обычно выполняются внутри графической библиотеки, а программисту приходится только указывать сведения, необходимые для проведения преобразований. Например, трансляции и повороты объектов учитываются при преобразовании модели, положение точки зрения, точки наблюдения и вектора вертикали – при преобразовании наблюдения, а тип проекции, расположение центра проекции и экрана – при преобразовании проектирования. Однако графические библиотеки примитивного уровня могут потребовать от программиста самостоятельного написания кода для всех этих преобразований.



Рис. 4.5. Преобразования между системами координат

4.1.2. Окно и видовой экран

Термин *окно* (window) в сетевой компьютерной среде обозначает область экрана монитора рабочей станции, посредством которого пользователь взаимодействует с вычислительными ресурсами, подключенными к той же сети. В компьютерной графике этот термин имеет иное значение. Окно – это область пространства, проецируемая на монитор. Объекты, находящиеся вне окна, на мониторе не появляются. В этом смысле оно подобно окну дома, через которое человеку, сидящему внутри дома, видна лишь часть внешнего мира. Вероятно, эта аналогия была основанием для выбора соответствующего термина. Окно обычно определяется как прямоугольник, лежащий на экране и заданный значениями X и Y . В системе координат просмотра (рис. 4.6, 4.7). Видимая область пространства, называемая *просматриваемым объемом* (viewing volume), зависит от типа проекции. Для параллельной проекции эта область имеет форму параллелепипеда, а для перспективной – форму пирамиды.

4.2. Примитивы в графических построениях

Примитивы (primitives) – это элементы графики, которые могут отображаться графической библиотекой. В каждой библиотеке набор примитивов свой, поэтому в данном разделе будут рассмотрены только наиболее общие примитивы, поддерживаемые большинством графических библиотек.

Отрезок. Для отображения *отрезка прямой* (линии – line) необходимо задание координат двух его концов. В большинстве графических библиотек координаты концов могут задаваться в трехмерном пространстве; проецирование на плоскость экрана осуществляется автоматически. Можно указывать атрибуты отрезка: тип, толщину, цвет и другие. Типы отрезков, поддерживаемых большинством графических библиотек, изображены на рис. 4.8. Для систем автоматизированной разработки чертежей поддержка этих типов линий совершенно необходима, поскольку они часто используются в машиностроительных и архитектурных чертежах и электрических схемах.



Рис. 4.8. Различные виды отрезков

В библиотеках GKS, PHIGS и OpenGL одной из базовых функций является *ломаная* (polyline), представляющая собой набор соединенных друг с другом отрезков. Координаты концов отрезков, составляющих ломаную, задаются в виде матрицы. В случае ломаной, состоящей всего из одного отрезка, в матрицу помещаются координаты двух его концов

$$[P] = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}.$$

Многоугольник – это то же самое, что и ломаная, за небольшим исключением: первая и последняя строки матрицы вершин $[P]$ должны быть одинаковыми (соответствующие им точки совпадают). Того же результата можно было бы достичь и с использованием функции построения ломаной, однако многоугольник, построенный при помощи специальной функции, распознается системой как объект, имеющий внутреннюю и внешнюю части. Внутренняя площадь многоугольника может быть заполнена штриховкой различных видов (рис. 4.9).

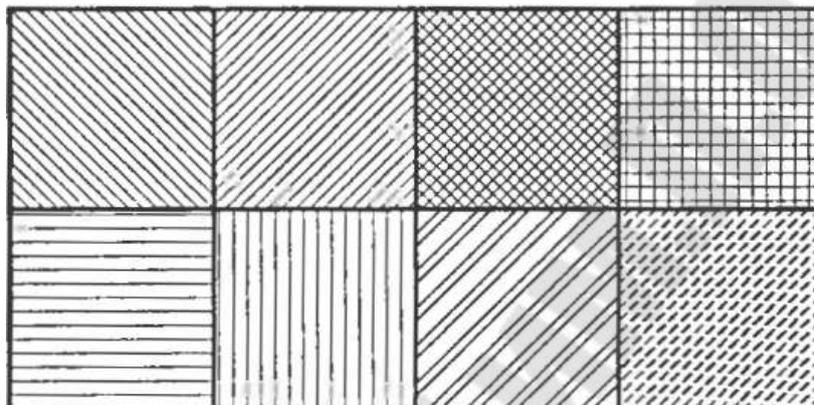


Рис. 4.9. Примеры различных заливок

Атрибутами многоугольника могут быть цвет внутренней области (цвет заполнения), а также тип, ширина и цвет ломаной, ограничивающей эту область. Хотя функция построения многоугольников может использоваться и для построения кругов и прямоугольников, в большинстве графических библиотек существуют специальные функции, требующие гораздо меньше входных параметров (например, центр и радиус круга или два конца диагонали прямоугольника). Тем не менее, внутри библиотек эти функции реализованы через функцию построения многоугольников.

Маркер. Маркеры обычно используются для выделения точек на графиках. Маркеры, доступные в большинстве графических библиотек, показаны на рис. 4.10. Тип маркера указывается в качестве атрибута. Полимаркер, как и отрезок, является стандартным объектом в GKS и PHIGS. OpenGL не поддерживает маркеры явно, однако предоставляет механизм сохранения маркеров в растровых файлах и вывода их на экран. Благодаря этому графическая программа, построенная на OpenGL, гораздо лучше переносится на различные платформы. Приведенный ниже образец кода демонстрирует вывод маркера * (звездочка) в PHIGS и OpenGL.



Рис. 4.10. Примеры маркеров

Текст. Большинство графических библиотек поддерживают два вида текста: текст для пояснений (экранный или двумерный текст) и трехмерный текст. Текст для пояснений всегда располагается в плоскости экрана, поэтому его форма не искажается вне зависимости от угла, на который он повернут. Трехмерный текст может быть расположен на любой плоскости в трехмерном пространстве. Его положение и ориентация задаются в мировых координатах. Для текста любого вида необходимо задание таких параметров, как шрифт, отношение высоты к ширине и угол наклона букв, а также положение и направление строки текста. Текст может быть представлен символами двух видов: аппаратными и программными. Программный шрифт строится соответствующими графическими программами, заранее сохраняемыми в памяти компьютера. Построение его занимает больше времени, чем построение символов аппаратного шрифта, но зато форма может быть гораздо более замысловатой. Символы аппаратного шрифта состоят из отдельных отрезков, формирующих буквы.

Ввод графики. Как уже отмечалось, графической программе может требоваться поддержка ввода графических элементов, таких как точки, отрезки и многоугольники, а не только чисел и текстовых строк. Например, если пользователь хочет вычислить площадь многоугольника на экране или увеличить его размеры, он должен сначала указать интересующий его многоугольник среди прочих объектов, видимых на экране. Для ввода графики используется два вида физических устройств: локатор (устройство ввода координат) и кнопка. *Локатор* (locator) передает графической программе информацию о своем положении, т. е. о положении курсора. *Кнопка* (button) сообщает о действиях пользователя (включении и выключении) в месте текущего положения курсора. В наши дни наиболее популярным устройством графического ввода является мышь, которая выполняет обе функции. Шарик в нижней части корпуса мыши позволяет вводить координаты, а кнопки наверху корпуса передают программе действия пользователя.

Устройство графического ввода может работать в трех режимах: опрос, запрос и выбор. В режиме *опроса* (sampling) осуществляется

постоянное считывание состояния устройства ввода, прежде всего положения локатора. Например, если вы свободно рисуете на экране, перемещая мышь, она работает в режиме опроса. Перемещение мыши приводит к непрерывному перемещению курсора по экрану. В режиме *запроса* (requesting) положение локатора считывается только при отправке запроса, которая обычно производится при нажатии на кнопку мыши. Чтобы прояснить различие между режимами опроса и запроса, рассмотрим процесс построения многоугольника путем графического задания координат его вершин при помощи мыши. В этом случае мы перемещаем мышь до тех пор, пока курсор не окажется в нужном месте, после чего нажимаем кнопку. Курсор перемещается по экрану согласно движениям мыши, которая находится при этом в режиме опроса. Координаты вершин передаются графической программе в режиме запроса. У этих режимов есть общее свойство: графической программе передаются координаты мыши или курсора. В режиме *выбора* (picking) устройство графического ввода идентифицирует элемент экрана, на который указывает курсор в момент нажатия кнопки. Графические элементы можно идентифицировать по именам, присвоенным им программистом во время составления программы. Режим выбора очень удобен при редактировании чертежа, уже имеющегося на экране (например, для удаления многоугольников или изменения координат их вершин).

Дисплейный файл. Дисплейный файл (display list) – это группа команд графической библиотеки, сохраненная для последующего выполнения. Большинство команд графических библиотек могут либо помещаться в дисплейный файл, либо выполняться немедленно. Дисплейный файл обеспечивает удобство и эффективность упорядочения и обработки команд библиотеки. Рассмотрим, например, перемещение изображения дома по экрану. Предположим, что рисунок состоит из нескольких сотен отрезков. Если бы эти отрезки существовали по отдельности, нам пришлось бы написать команду перемещения несколько сотен раз – для каждого из них. Однако если команды построения отрезков, образующих рисунок, объединены в дисплейный файл, команду перемещения достаточно написать только один раз. Чтобы поместить графические элементы в дисплейный файл, нужно открыть этот файл перед первой командой, которая должна в него попасть, и закрыть его после последней команды.

Дисплейный файл OpenGL ориентирован на оптимизацию производительности, в частности, при работе по сети, но не за счет про-

производительности на отдельном компьютере. Оптимизация обеспечивается благодаря тому, что дисплейный файл хранится в виде списка команд, а не в виде динамической базы данных. Другими словами, созданный дисплейный файл изменить уже нельзя. Если бы его можно было изменять, производительность упала бы из-за накладных расходов на поиск внутри списка и управление памятью. Изменение отдельных частей дисплейного файла потребовало бы перераспределения памяти, что могло бы привести к ее фрагментации. Дисплейные файлы, как правило, работают так же быстро, как и обычные последовательности команд, не объединенные в группы. В случае OpenGL дисплейные файлы могут значительно повысить производительность, в особенности при передаче подпрограмм OpenGL по сети, поскольку файлы эти хранятся на сервере, благодаря чему сокращается сетевой трафик. К созданному дисплейному файлу могут быть применены следующие операции:

- множественное выполнение (multiple execution) – один и тот же файл можно выполнять много раз;

- иерархическое выполнение (hierarchical execution) – иерархическим называется дисплейный файл (родительский), вызывающий другие дисплейные файлы (дочерние). Иерархические дисплейные файлы удобны для объектов, состоящих из отдельных компонентов, особенно, если некоторые компоненты входят в объект в нескольких экземплярах;

- удаление (deletion) – дисплейный файл может быть удален.

Матрица преобразования. Проецирование точек на объект в трехмерном пространстве требует преобразования координат из одной системы в другую. Сначала нужно перевести координаты точек объекта из модельной системы в мировую. Текущее положение объекта обычно задается через повороты и смещения относительно исходного положения, в котором модельная система координат совпала с мировой. Следовательно, мировые координаты точек объекта можно получить трансляцией и поворотом соответствующих точек из их исходного положения, в котором их модельные координаты совпали с мировыми. Большинство графических библиотек выполняют эти преобразования самостоятельно, а программисту остается задать только смещение и поворот для интересующего его объекта. Однако проектировщику все равно нужно знать законы преобразований, чтобы рисовать объекты в нужных местах без проб и ошибок, в особенности, если эти объекты перемещаются достаточно сложным образом.

В этом разделе рассматриваются матрицы преобразования, действующие на точки объекта при его перемещении и повороте.

Получив мировые координаты всех точек объекта в его текущем положении, мы должны вычислить координаты этих точек в наблюдательской системе. Перевод координат из одной системы в другую называется *отображением* (mapping). Отображение между мировой и наблюдательской системами координат обычно также осуществляется графической библиотекой самостоятельно, по заданным программистом координатам точки зрения, точки наблюдения и направлению вектора вертикали (в мировых координатах).

4.3. Системы автоматизированной разработки чертежей

Система автоматизированной разработки чертежей (computer-aided drafting system) – это программный продукт, позволяющий разработчику в интерактивном режиме создавать и изменять машиностроительные, архитектурные, инженерные чертежи, электрические схемы и чертежи множества других разновидностей. Эта программа, кроме того, обновляет базу данных, сохраняя готовые чертежи и их изменения. Таким образом, работа с системой автоматизированной разработки чертежей аналогична использованию текстового процессора. Единственное отличие в том, что на выходе пользователь получает чертеж, а не текстовый документ. Как в текстовом процессоре можно очень быстро подготовить новый документ на базе существующего, так и в системе автоматизированной разработки чертежей можно получить новый чертеж, изменив имеющиеся. Преимущества текстового процессора или автоматизированной системы разработки чертежей трудно оценить при подготовке абсолютно нового документа или чертежа. Но при изменении существующих документов и чертежей их преимущества становятся очевидными и неопределимыми.

Настройка параметров чертежа. Работу с системой автоматизированной разработки чертежей следует начинать с установки параметров, таких как единицы измерения, размеры чертежа, параметры сетки и слоев. Для быстрого и точного построения чертежей необходимо, чтобы все эти параметры имели правильные значения. Чертеж можно построить без сетки и без слоев, но на это уйдет много времени, а изменить получившийся чертеж будет очень сложно.

Единицы измерения. Пользователь должен выбрать формат и точность единиц измерения расстояний и углов. Единицы измерения

расстояний могут быть представлены в научном, десятичном, дробном, инженерном и архитектурном форматах. Единицы измерения углов – это градусы, градусы/минуты/секунды, грады, радианы и геодезические единицы.

Размеры чертежа. Рисуя чертеж на бумаге, вы не можете выйти за границы листа. Точно так же и при работе с графическим устройством чертеж должен иметь определенные границы, поскольку этот чертеж когда-нибудь все равно будет напечатан на бумаге конечного размера. Значит, пользователь должен заранее установить размеры чертежа.

Command: limits

При выборе размеров чертежа обычно учитывают следующие факторы:

- фактический размер чертежа;
- пространство для нанесения размеров, примечаний, списков материалов и других необходимых данных;
- расстояние между разными видами (чертеж не должен выглядеть загроможденным);
- пространство для рамки и заголовка, если таковые предусмотрены.

Перед тем как задавать размеры чертежа, рекомендуется построить его эскиз, чтобы грубо оценить необходимое пространство. Например, если размеры вида спереди для какого-то объекта равны 6×5 единиц, размеры вида сбоку – 4×5 единиц и размеры вида сверху – 6×4 единиц, ограничения должны быть установлены таким образом, чтобы вместить весь чертеж и все относящиеся к нему данные. Предположим, необходимо, чтобы расстояние между видами спереди и сбоку составляло 4 единицы, а расстояние между видами спереди и сверху – 3 единицы. Расстояние до границ чертежа пусть составляет 4 единицы слева, 4 справа, 2 сверху и 2 снизу. Указанные значения выбираются таким образом, чтобы готовый чертеж выглядел гармонично.

Установив для себя размеры видов и расстояния между ними, а также расстояния до границ чертежа и от границ до краев бумаги, можно вычислить размеры чертежа следующим образом:

- размер по горизонтали = $1 + 4 + 6 + 4 + 4 + 4 + 1 = 24$;
- размер по вертикали = $1 + 2 + 5 + 3 + 4 + 2 + 1 = 18$.

Итак, размеры чертежа составляют 24×18 единиц.

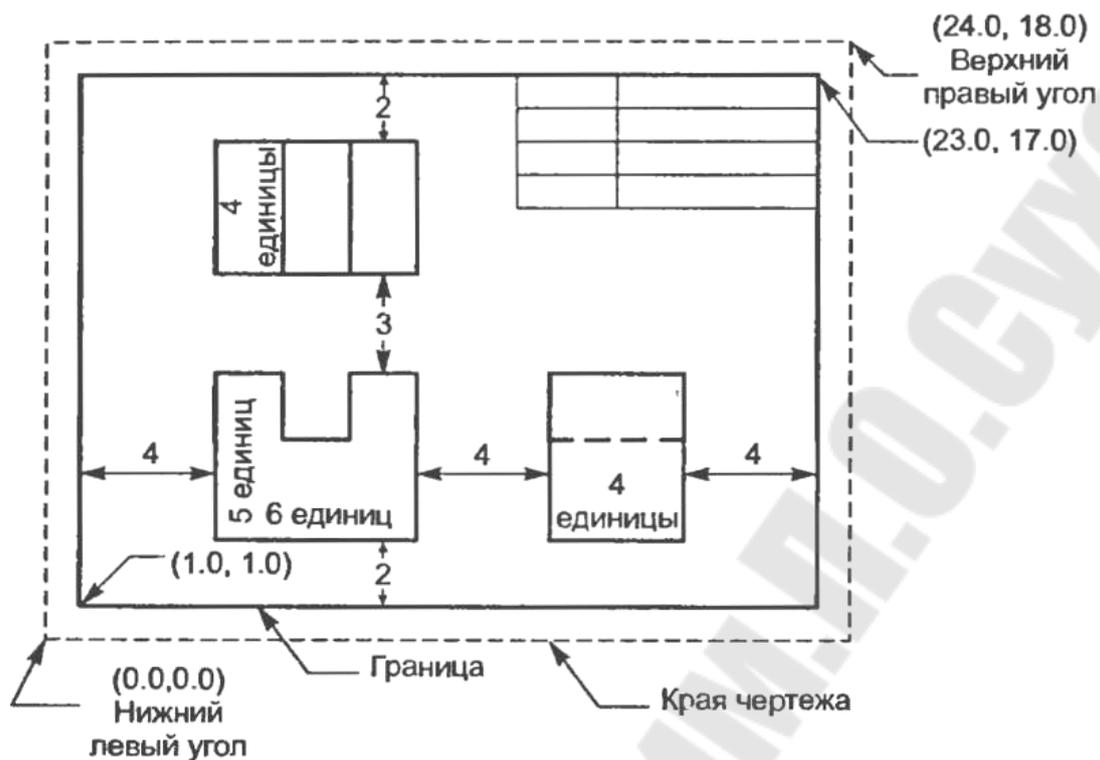


Рис. 4.11. Задание размеров чертежа

Чертеж на рис. 4.11 выполнен в масштабе 1:1. Однако, если вы хотите распечатать чертеж, чтобы получить твердую копию, вам придется увеличить или уменьшить его в зависимости от размеров листа. Вообще говоря, размер листа определяет границы чертежа, размер шрифта, масштаб чертежа, масштаб толщины линии и другие параметры чертежа.

Слой. Разбивать чертеж на множество слоев очень удобно, особенно если чертеж достаточно сложный. Распределив поэтажный план здания и схему трубопроводов по отдельным слоям, вы значительно упростите себе задачу. Другими словами, выполнять все операции с отдельным слоем значительно проще, чем с большим чертежом, содержащим все объекты. Однако надо иметь возможность переключаться между режимами просмотра, чтобы получить представление об относительном расположении элементов из разных слоев (например, труб и стен здания). Разделение на слои сохраняет возможность накладывать их друг на друга, не усложняя при этом чертеж, по крайней мере, в том, что касается выполняемых графических операций. Слой, с которым вы работаете в данный момент, считается активным, тогда как все остальные слои считаются неактивными. Как и любой другой фон, графические элементы неактивных слоев нечувствительны к графическим операциям, таким как выбор

или удаление. Поэтому сложность чертежа остается той же самой, как если бы вы работали с одним-единственным активным слоем.

Функция разделения на слои может эффективно использоваться при построении чертежей отдельных слоев многослойных печатных плат. В этом случае каждый слой может строиться независимо от остальных, однако остается возможность вывести на экран связанные каким-либо образом слои для получения сведений об их относительном положении. Разделение на слои удобно и для построения чертежей отдельных деталей конструкции. Если рисовать каждую деталь в отдельном слое сборочного чертежа, чертеж любой детали легко будет получить, сделав активным нужный слой.

Сетка и привязка. В черчении на бумаге широко используются вспомогательные линии, которые строятся заранее при помощи рейсшины. Они облегчают построение линий чертежа и делают их более точными. Линии сетки в системах автоматизированной разработки чертежей имеют то же назначение, что и линии построения в черчении. Горизонтальные и вертикальные линии сетки рисуются на равных расстояниях друг от друга в соответствии с заданным разрешением, а линии чертежа строятся поверх них. В некоторых системах автоматизированной разработки чертежей строятся только точки на перекрестьях линий сетки.

Чтобы провести прямую линию поверх линии сетки, нужно задать положение двух ее концов. Их координаты можно ввести с клавиатуры или указать мышью, установив курсор в нужное положение и нажав кнопку. Курсор отслеживает движение мыши, когда она находится в режиме локатора. Положение точки, указанное вторым методом, может быть не совсем точным из-за дрожания человеческой руки или неточности механизма мыши. Чтобы справиться с этой проблемой, можно включить привязку курсора к ближайшему пересечению линий сетки. При нажатии кнопки мыши компьютер будет воспринимать точные координаты этого пересечения. Точность задания координат будет определяться разрешением сетки, которое пользователь может настраивать по своему желанию. Эта функция называется *привязкой* (snapping).

Слайн. Слайны используются для построения произвольных кривых подобно тому, как в черчении от руки это делается с помощью лекала. Пользователь указывает точки на кривой, а система строит интерполяционную кривую, проходящую через эти точки. Получившаяся кривая обычно представляется уравнением третьего по-

рядка. Иногда кривые могут строиться по задающим точкам, которые определяют кривую, но не обязаны лежать на ней.

Простановка размеров. Возможность простановки размеров считается одной из наиболее привлекательных особенностей систем автоматизированной разработки чертежей. Вручную проставить размеры с той же быстротой просто невозможно. В системах автоматизированной разработки чертежей простановка размеров осуществляется следующим образом. Чтобы указать горизонтальный или вертикальный размер, достаточно всего лишь выбрать два графических элемента (обычно точки) и желаемое положение размерной линии. В этом случае расстояние между точками автоматически определяется по чертежу. Стрелки, размерные линии, выносные линии и значение размера наносятся системой самостоятельно. Система автоматически измеряет расстояние по вертикали, если графические элементы расположены на вертикальной линии, или расстояние по горизонтали, если они находятся на горизонтальной линии. Если объекты расположены как-то иначе, система просит уточнить, какой именно размер вы хотите проставить: вертикальный, горизонтальный или реальный.

Размеры радиусов и диаметров проставляются путем выбора окружности или дуги и последующего указания положения размерной линии. Угловые размеры проставляются аналогичным образом: нужно выбрать два отрезка и указать положение размерного текста. Какой именно угол будет измерен (внешний или внутренний), зависит от порядка выбора отрезков. В каждой системе используется свое собственное соглашение о порядке выбора.

Может возникнуть вопрос: зачем проставлять размеры в интерактивном режиме, если чертеж уже содержит все сведения о размерах и положении объектов? Теоретически можно было бы полностью автоматизировать простановку размеров на чертежах, но на практике разработчики сталкиваются со следующими проблемами. Существует много способов проставить размеры на одном и том же чертеже. Вообще говоря, проектировщик учитывает методы производства, контроля и сборки детали, изображенной на чертеже, и на основании своего опыта выбирает оптимальную схему образмеривания. Воспроизвести опыт проектировщика на компьютере очень сложно. Более того, найти хотя бы одну полную и безызыточную схему расстановки размеров не так-то просто. В настоящий момент ведутся исследования возможности решения этой задачи.

Примечания. Чтобы добавить к чертежу примечание, т. е. текстовую строку, нужно задать расположение и ориентацию этой строки, а также размер и шрифт символов. Последние три параметра обычно имеют некоторые значения по умолчанию, которые используются в том случае, если пользователь не указывает никаких конкретных значений.

Макропрограммирование. Программирование макросов или макропрограммирование заключается в объединении наборов графических команд под одним именем. Если графические команды объединяются в программу, которая называется *макропрограммой* (macro program), к ним могут добавляться некоторые условные и арифметические операторы из обычных компьютерных языков. Входные параметры графических команд могут быть определены как переменные, что позволяет задавать макропрограмме разные значения и получать разные чертежи. Макропрограмма такого рода называется *параметрической программой* (parametric program), поскольку чертеж, который она строит, зависит от значений, присвоенных соответствующим параметрам. Хорошим примером параметрической программы может быть автоматическая программа построения чертежей винтов. Пользователь вводит характеристики нагрузки, программа рассчитывает размеры винтов по этим характеристикам, а затем строит их чертежи с учетом вычисленных размеров. В такую параметрическую программу входят арифметические операторы, позволяющие вычислить размеры винтов, а также графические команды, строящие чертеж винта. Функция макропрограммирования очень важна, поскольку она позволяет приспособлять коммерческие системы автоматизированной разработки чертежей под конкретные приложения. Многообразие параметрических программ, разработанных компанией, фактически может быть мерой эффективности использования этой компанией имеющейся у нее системы автоматизированной разработки чертежей.

Измерения. Функция измерения позволяет выполнять вычисления по готовому или строящемуся чертежу. Система позволяет определить площадь любой области, угол между двумя отрезками, минимальное расстояние между графическими элементами и другие параметры. Эта функция может быть очень полезна, если построение чертежа и проектирование осуществляются в системе автоматизированной разработки чертежей одновременно. Например, проектировщик может проверить, соответствует ли получившаяся конструкция требованиям к площади теплопередачи или к минимальному пространству для обслуживания.

Дополнительные функции. Помимо функций, описанных выше, имеются вспомогательные функции, позволяющие обновлять чертежи в базе данных, получать чертежи из базы данных и составлять списки материалов.

Совместимость файлов чертежей. Настоящим преимуществом системы автоматизированной разработки чертежей является возможность сохранения файла чертежа в базе данных, откуда его смогут получить сотрудники разных отделов. Этим достоинством легко воспользоваться на практике, если все сотрудники работают в одной системе автоматизированной разработки чертежей и не испытывают проблем с чтением файлов, сделанных другими сотрудниками. Однако преимущество легко утратить, если разные отделы в одной и той же компании работают с разными системами, не способными читать файлы друг друга. Проблема становится еще более серьезной, если речь идет о системах разных производителей. В этом случае единственным реальным методом взаимодействия будет построение чертежей на бумаге и механическое их воспроизведение подобно тому, как это делалось в прошлом.

Чтобы избежать этой проблемы, можно потребовать от всех производителей систем автоматизированной разработки чертежей сохранения файлов в стандартном формате. Практически все коммерческие системы автоматизированной разработки чертежей поддерживают формат IGES.

5. СИСТЕМЫ ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ

Процесс проектирования может рассматриваться как постепенная детализация формы по мере развития идей разработчика. Программное обеспечение автоматизированного проектирования – это просто одно из средств, облегчающих этот процесс. Системы геометрического моделирования позволяют работать с формами в трехмерном пространстве.

Системы геометрического моделирования были созданы для того, чтобы преодолеть проблемы, связанные с использованием физических моделей в процессе проектирования. Эти системы создают среду, подобную той, в которой создаются и изменяются физические модели. Другими словами, в системе геометрического моделирования разработчик изменяет форму модели, добавляет и удаляет ее части, детализируя форму визуальной модели таким же образом, как ребенок формирует фигурку из пластилина. Визуальная модель может выглядеть точно так же, как физическая, но она нематериальна. Однако трехмерная визуальная модель хранится в компьютере вместе со своим математическим описанием, благодаря чему устраняется главный недостаток физической модели – необходимость выполнения измерений для последующего прототипирования или серийного производства. Системы геометрического моделирования делятся на каркасные, поверхностные, твердотельные и немногочисленные (перечислены в порядке эволюционирования).

5.1. Системы каркасного моделирования

В *системах каркасного моделирования* (wireframe modeling systems) форма представляется в виде набора характеризующих ее линий и конечных точек. Линии и точки используются для представления трехмерных объектов на экране, а изменение формы осуществляется путем изменения положения и размеров отрезков и точек. Другими словами, визуальная модель представляет собой каркасный чертеж формы, а соответствующее математическое описание представляет собой набор уравнений кривых, координат точек и сведений о связности кривых и точек. Сведения о связности описывают принадлежность точек к конкретным кривым, а также пересечение кривых друг с другом. Системы каркасного моделирования были популярны в ту пору, когда геометрическое моделирование только начало зарождаться. Их популярность объяснялась тем, что в системах каркасного

моделирования создание форм выполнялось через последовательность простых действий, так что пользователям было достаточно легко создавать формы самостоятельно. Однако визуальная модель, состоящая из одних лишь линий, может быть неоднозначной (рис. 5.1).

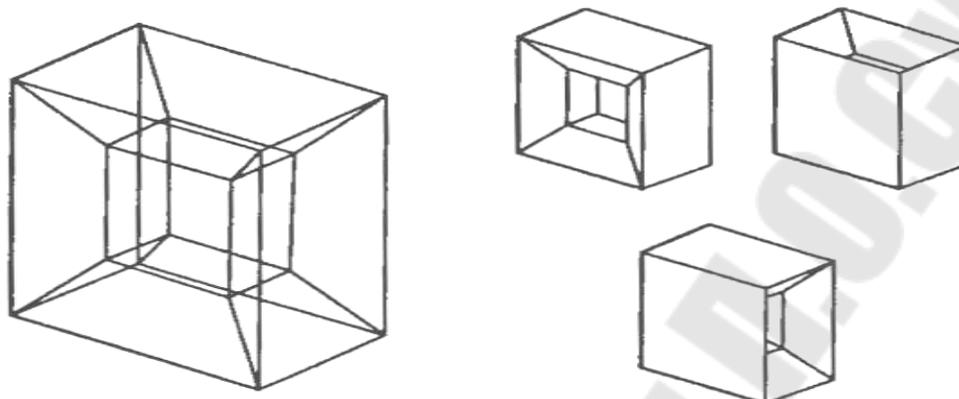


Рис. 5.1. Неоднозначные каркасные модели

Более того, соответствующее математическое описание не содержит сведений о внутренних и внешних поверхностях моделируемого объекта. Без этих сведений невозможно рассчитать массу объекта, определить траектории перемещения инструмента при обработке объекта или создать сетку для конечноэлементного анализа несмотря на то, что объект кажется трехмерным. Поскольку эти операции являются неотъемлемой частью процесса проектирования, системы каркасного моделирования были постепенно вытеснены системами поверхностного и твердотельного моделирования.

5.2. Системы поверхностного моделирования

В системах поверхностного моделирования (surface modeling systems) математическое описание визуальной модели включает в себя не только сведения о характеристических линиях и их конечных точках, как в каркасном моделировании, но и данные о поверхностях. При работе с отображаемой на экране моделью изменяются уравнения поверхностей, уравнения кривых и координаты конечных точек. Если поверхности не окрашены и не затушеваны, визуальная модель в системе поверхностного моделирования может выглядеть точно так же, как в системе каркасного моделирования.

Математическое описание может включать сведения о связности поверхностей, т. е. о том, как поверхности соединяются друг с другом и по каким кривым. Однако в математическое описание моделей, соз-

дававшихся в системах поверхностного моделирования, обычно включались только уравнения бесконечных поверхностей (или их параметры) без сведений о связности.

Примерами атрибутов, определяющих поверхность, являются положение и направление центральной оси, и радиус цилиндра. Из-за отсутствия сведений о связности приложению типа программы для формирования траекторий перемещения приходилось определять границы поверхностей и проверять их связность самостоятельно. В современных системах поверхностного моделирования такие неудобства исключаются благодаря включению сведений о связности поверхностей.

Существует три стандартных метода создания поверхностей в системах поверхностного моделирования.

1. Интерполяция входных точек.
2. Интерполяция криволинейных сеток.
3. Трансляция или вращение заданной кривой.

Способы ввода для каждого метода могут зависеть от конкретной системы поверхностного моделирования. Однако базовый метод ввода для каждой системы легко определить по представлению кривых и плоскостей.

Системы поверхностного моделирования используются для создания моделей со сложными поверхностями, потому что визуальная модель позволяет оценить эстетичность проекта, а математическое описание позволяет построить программу для обработки поверхностей детали на станке с числовым программным управлением (ЧПУ). Расчет и проверку траектории движения фрезы с ЧПУ для объекта, созданного в системе поверхностного моделирования, демонстрирует рис. 5.2.



Рис. 5.2. Расчет и проверка траектории станка с ЧПУ (программа HyperMILL)

5.3. Системы твердотельного моделирования

Системы твердотельного моделирования (solid modeling systems) предназначены для работы с объектами, состоящими из замкнутого объема, или *монолита* (solid). В системах твердотельного моделирования, в отличие от систем каркасного и поверхностного моделирования, не допускается создание наборов поверхностей или характеристических линий, если они не образуют замкнутого объема. Математическое описание объекта, созданного в системе твердотельного моделирования, содержит сведения, по которым система может определить, где находится какая-либо точка: внутри объема, снаружи него или на его границе. По этим сведениям можно получить любую информацию об объеме тела, а значит, могут быть созданы приложения, работающие с объектом на уровне объема, а не на базе поверхности.

Например, можно написать приложение, формирующее сетку конечных элементов объемного типа по твердотельной модели. Можно написать программу для формирования всех траекторий фрезы с ЧПУ, необходимых для изготовления детали из заготовки. Эта программа будет работать не с отдельными плоскостями, а с объемом в целом, что исключит потребность во вводе данных для каждой плоскости. Все эти возможности реализуются в том случае, если модель создается в виде замкнутого объема. Однако создание модели в виде замкнутого объема требует большего количества входных данных по сравнению с количеством данных, дающих математическое описание. Это одна из причин, по которым были разработаны системы моделирования немногочисленных объектов. Такие системы моделирования позволяют работать с поверхностями и замкнутыми объемами одновременно.

Если бы система твердотельного моделирования требовала ввода всех данных для полного математического описания, она была бы слишком сложной для пользователей, и они отказались бы от нее. Процесс детализации формы не был бы похож на интуитивный процесс физического моделирования, и в результате получилось бы совсем не то, на что рассчитывали создатели систем геометрического моделирования. Поэтому разработчики систем твердотельного моделирования стараются предоставить простые и естественные функции, чтобы пользователи могли работать с объемными формами точно так же, как они работают с физическими моделями, не вдаваясь в подробности математического описания. Функции моделирования, подобные созданию примитивов, булевым операциям, поднятию, построению

фигуры вращения, повороту и закруглению, требуют от пользователя совсем немного. Обо всех деталях математического описания системы заботятся сами.

5.3.1. Функции моделирования

Функции моделирования, поддерживаемые большинством систем твердотельного моделирования, могут быть разделены на пять основных групп. В первую группу входят функции, используемые для создания простых форм на основе объемных заготовок, имеющих в программе, – так называемые *функции создания примитивов* (primitive creation functions). К этой же группе относятся функции добавления и вычитания объема – *булевские операторы* (Boolean operations). Функции моделирования из первой группы позволяют проектировщику быстро создать форму, близкую к окончательной форме детали.

Ко второй группе относятся функции создания объемных тел путем перемещения поверхности. Функция заметания (sweeping) позволяет создавать объемное тело трансляцией или вращением области, заданной на плоскости. Построение тела вращения из плоской кривой называется также качанием или вращательным заметанием (swinging). Задавая замкнутую плоскую область, пользователь может указывать геометрические ограничения или вводить данные о размерах, а не рисовать форму вручную. Здесь под геометрическими ограничениями понимаются соотношения между элементами рисунка (перпендикулярность отрезков, касание дуги окружности отрезком и т. д.). В этом случае система построит точную форму, удовлетворяющую ограничениям, самостоятельно. Изменение геометрических ограничений или размеров даст другую плоскую область и другое объемное тело. Такой подход называется параметрическим моделированием, поскольку изменение параметров позволяет получить разные объекты. Параметрами могут быть постоянные, входящие в геометрические ограничения, а также размеры. Функция *скиннинга* (skinning) создает объемное тело, натягивая поверхность на заданные поперечные сечения. Функции второй группы позволяют проектировщику начать моделирование с формы, весьма близкой к конечному результату, поскольку одних поперечных сечений вполне достаточно для точного описания конечного объемного тела.

В третью группу входят функции моделирования, предназначенные главным образом для изменения существующей формы.

Типичными примерами являются функции *скругления* или *плавного сопряжения* (rounding, blending) и *поднятия* (lifting). К четвертой группе относятся функции, позволяющие непосредственно манипулировать составляющими объемных тел, т. е. вершинами, ребрами и гранями. Работа с этими функциями (аналогичными функциям систем поверхностного моделирования) называется *моделированием границ* (boundary modeling). В последнюю группу входят функции, используя которые проектировщик может моделировать твердое тело при помощи свободных форм. Например, он может давать системе команды типа «сделать отверстие такого-то размера в таком-то месте» или «сделать фаску такого-то размера в таком-то месте». Работа с такими функциями называется *объектно-ориентированным моделированием* (feature-based modeling). В последнее время функциям пятой группы уделяется особое внимание, поскольку модель, построенная с их помощью, содержит информацию о процессе создания, без которой невозможно автоматическое формирование плана технологического процесса для детали. Заметьте, что модель, созданная другими средствами, содержит только элементарные геометрические сведения о вершинах, ребрах и гранях.

Функции создания примитивов. Функции создания примитивов позволяют выбирать и создавать простейшие объекты, заранее определенные авторами системы моделирования. Размер примитива задается пользователем.

Булевские операции. Если бы в списке примитивов можно было найти любое объемное тело, это было бы замечательно. Однако из-за разнообразия возможных применений систем геометрического моделирования сохранить заранее все мыслимые формы невозможно. Гораздо проще приблизиться к решению, предоставив пользователю средства для комбинирования примитивов. В качестве метода комбинирования в твердотельном моделировании применяются булевские операции теории множеств. Другими словами, каждое примитивное объемное тело считается множеством точек, к множествам применяются булевские операции, а в результате получается объемное тело, состоящее из точек, полученных после преобразований.

Большинством систем твердотельного моделирования поддерживаются следующие булевские операции: объединение, пересечение и разность. До применения булевских операций необходимо определить относительное положение и ориентацию примитивов. Булевские операции могут применяться не только к примитивам, хотя на рисунках в качестве примеров изображены именно примитивы.

Моделирование границ. Функции *моделирования границ* используются для добавления, удаления и изменения элементов объемного тела – его вершин, ребер и граней. Следовательно, процедура, использующая функции *моделирования границ*, будет выглядеть точно так же, как в системах *поверхностного моделирования*. Другими словами, вначале создаются точки, затем создаются ребра, соединяющие эти точки, и наконец, граничные ребра определяют поверхность. Однако в системах *твердотельного моделирования*, в отличие от систем *поверхностного моделирования*, нужно определить все поверхности таким образом, чтобы образовался замкнутый объем. Создание клина при помощи функций *моделирования границ* показано на рис. 5.3. Процедура включает создание точек, граней и поверхностей.

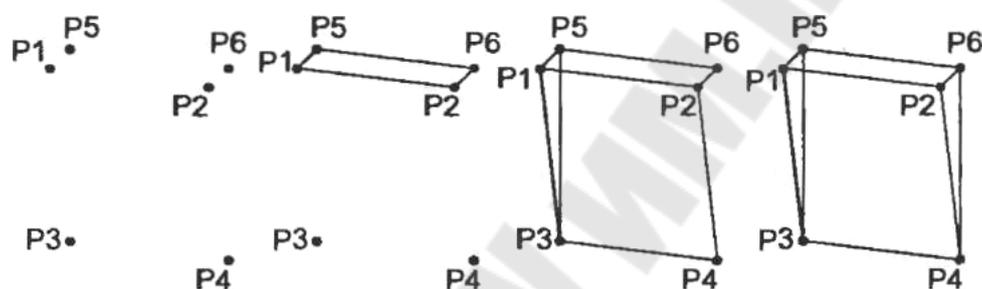


Рис. 5.3. Создание тела при помощи функций моделирования границ

Создавать объемное тело исключительно при помощи функций *моделирования границ* очень утомительно. Эти функции используются главным образом для создания плоских фигур, которые затем служат сечениями объемным телам, образуемым заметанием или *скиннингом*. Однако функции *моделирования границ* удобно применять для изменения формы уже существующего тела. Вершину можно передвинуть в новое положение, изменив соответствующим образом соседние ребра и грани (рис. 5.4). Прямое ребро можно заменить криволинейным, в результате чего изменятся связанные грани и вершины (рис. 5.4). Плоскую поверхность можно заменить на криволинейную с модификацией ребер и вершин (рис. 5.5). Иногда плоскую поверхность можно заменять криволинейной, импортированной из системы *поверхностного моделирования*. Перечисленные функции *моделирования* называются *функциями тонкого редактирования* (*tweaking functions*). Они используются для моделирования тел, ограниченных криволинейными поверхностями, поскольку такие поверхности легко получать из плоских граней многогранника.

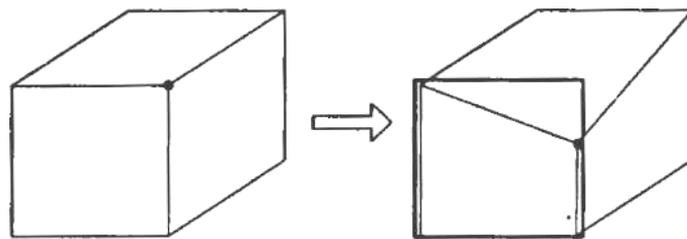


Рис. 5.4. Модификация объекта перемещением вершины

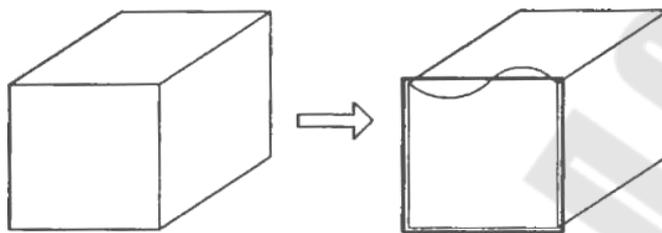


Рис. 5.5. Модификация объекта заменой ребра

5.3.2. Объектно-ориентированное моделирование

Объектно-ориентированное моделирование (feature-based modeling) позволяет конструктору создавать объемные тела, используя привычные *элементы форм* (features). Созданное тело несет в себе информацию об этих элементах в дополнение к информации об обычных геометрических элементах (вершинах, ребрах, гранях и др.). Например, конструктор может давать команды типа «сделать отверстие такого-то размера в таком-то месте» или «сделать фаску такого-то размера в таком-то месте», и получившаяся фигура будет содержать сведения о наличии в конкретном месте отверстия (или фаски) конкретного размера. Набор доступных в конкретной программе элементов формы зависит от спектра применения этой программы.

Большинством систем объектно-ориентированного моделирования поддерживаются такие элементы, которые используются при изготовлении деталей: фаски, отверстия, скругления, пазы, выемки и т. д. Такие элементы называются производственными, поскольку каждый из них может быть получен в результате конкретного процесса производства. Например, отверстие создается сверлением, а выемка – фрезерованием. Следовательно, на основании сведений о наличии, размере и расположении производственных элементов можно попытаться автоматически сформировать план технологического процесса. Автоматическое планирование технологического процесса, если оно будет разработано на практическом уровне, перебросит мост

между системами CAD и CAM, которые в настоящий момент существуют отдельно друг от друга.

Один из недостатков объектно-ориентированного моделирования заключается в том, что система не может предоставить все элементы, нужные для всех возможных приложений. Для каждой задачи может потребоваться свой набор элементов. Чтобы исключить этот недостаток, большинство систем объектно-ориентированного моделирования поддерживают какой-либо язык, на котором пользователь при необходимости может определять свои собственные элементы. После определения элемента необходимо задать параметры, указывающие его размер. Элементы, как и примитивы, могут быть разного размера, а задаются размеры параметрами в момент создания элемента. Создание элементов разного размера путем присваивания различных значений соответствующим параметрам является разновидностью параметрического моделирования.

5.3.3. Параметрическое моделирование

Параметрическое моделирование (parametric modeling) заключается в том, что конструктор определяет форму заданием геометрических ограничений и некоторых размерных параметров. Геометрические ограничения описывают отношения геометрических элементов. Примерами ограничений являются параллельность двух граней, компланарность двух ребер, касательность криволинейного ребра к соседнему прямому и т. д. К размерным данным относятся не только заданные размеры формы, но и соотношения между размерами. Соотношения записываются конструктором в виде математических уравнений. Таким образом, параметрическое моделирование заключается в построении формы путем решения уравнений, выражающих геометрические ограничения, и уравнений, описывающих заданные размеры и соотношения между ними.

В параметрическом моделировании построение формы обычно осуществляется в приведенной ниже последовательности.

1. Строится грубый набросок плоской фигуры.
2. В интерактивном режиме вводятся геометрические ограничения и данные о размерах.
3. Строится плоская фигура, отвечающая ограничениям и требованиям к размерам.
4. Шаги 2 и 3 повторяются с изменением ограничений или размеров до тех пор, пока не будет получена нужная модель (рис. 5.6).

5. Объемное тело создается заметанием или вращением плоской фигуры. Толщина и угол поворота также могут стать размерными параметрами, что позволит при необходимости легко изменить созданную трехмерную форму.

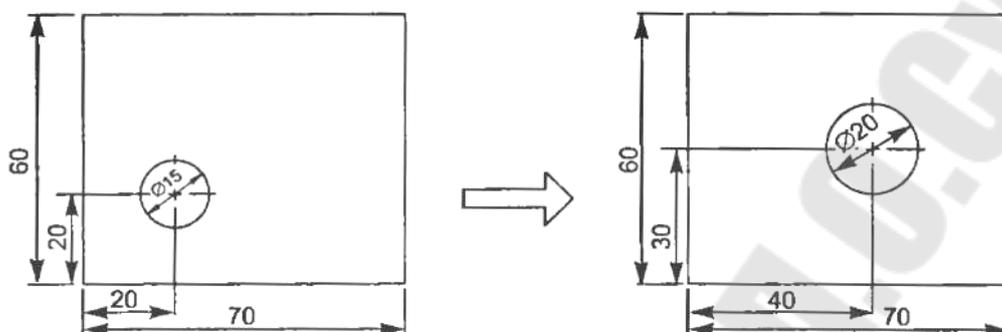


Рис. 5.6. Изменение формы через ограничения

Форма в параметрическом моделировании изменяется не непосредственно, а через использование геометрических ограничений и размерных соотношений. Поэтому конструктор может разработать множество альтернативных проектов, не заботясь о деталях, но сосредоточившись на функциональных аспектах.

Разновидности параметрического моделирования выделяются по методам решения уравнений, описывающих геометрические ограничения. Системы первого типа решают уравнения последовательно, а системы второго типа – одновременно. В первом случае форма тела зависит от порядка введения ограничений. Во втором случае, одна и та же форма получается независимо от последовательности введения ограничений, но в случае их конфликта могут возникнуть проблемы.

6. PLM-ТЕХНОЛОГИИ. ИНТЕГРАЦИЯ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

В последние несколько лет в мире САПР все чаще используется понятие PLM (ProductLifecycleManagement, в переводе – управление жизненным циклом изделия), отчасти заменившее уже привычные CAD, CAM, CAE и русскоязычную аббревиатуру САПР.

Некоторые поставщики решений автоматизации инженерных процессов, в первую очередь главные идеологи этой технологии Dassault Systems и Siemens PLM Software (бывшая Unigraphics PLM Solutions), используют только этот термин.

Существует распространенное мнение, что PLM – это просто новое название для комплекса программных средств проектирования (CAD), подготовки производства на ЧПУ (CAM) и инженерных расчетов (CAE), объединенных воедино системой управления документооборотом (PDM). Доля истины, конечно, в этом есть, все эти компоненты – неотъемлемая часть комплекса PLM, но лишь часть, хотя и ключевая. В целом же PLM – совокупность этих программных систем, методик их применения, а главное, людей, обладающих должной компетентностью. Можно сказать, это целая философия жизни производства, технически опирающаяся на программные средства. Фундаментом PLM являются три краеугольных камня:

- цифровые модели-прототипы изделия;
- модели процессов, связанных с его жизненным циклом – производством, эксплуатацией, обслуживанием, ремонтом и, наконец, утилизацией;
- модели ресурсов, вовлеченных во все стадии жизненного цикла продукта.

С момента появления технология PLM успешно внедрена на сотнях предприятий по всему миру. Тем самым она доказывает свое право на существование и зарекомендовала себя как мощный инструмент повышения эффективности производства и бизнеса в целом.

Крупнейшим и наиболее сложным с точки зрения масштаба и комплекса решаемых задач внедрение и использование PLM в АНПК «Сухой» (рис. 6.1). Система охватывает несколько КБ, серийных заводов и десятки смежников, многие из которых – зарубежные. В силу ряда причин АНПК «Сухой» одновременно использует программные решения от двух конкурирующих поставщиков: Siemens PLM Software (для проектирования и производства боевых самолетов) и Dassault Systemes (для создания гражданской авиатехники).

По словам начальника отдела САПР «ОКБ Сухого» Е. И. Савченко, на первоначальном этапе внедрения PLM-технологий под результатом предприятие понимало только выпуск бумажной документации, даже несмотря на то, что используются современные САД-системы, наращиваются производственные мощности, проходят обучение люди. Главный вывод из этого этапа – необходимо специально организовывать коллективную работу над проектом.

Следующий этап начался после построения компьютерной сети предприятия и удвоения числа рабочих мест. Были разработаны первые положения, регламентирующие организацию работ, и выполнены пилотные проекты по созданию электронного макета изделия под управлением PDM-системы Teamcenter Engineering. Целью было создание проектов легкомоторных самолетов. На этом этапе были отработаны основные принципы организации работ над электронным макетом. В их числе: ведение проекта «сверху вниз», распределение ответственности и прав доступа, система обозначений, стандартизация рабочего пространства пользователей в Teamcenter, отделение чертежа от модели, уточнение требований к моделям.

Наиболее важные нововведения были связаны с появлением технологии WAVE, позволяющей организовать управляемые ассоциативные связи между моделями. Было введено понятие «базовая контрольная структура» (БКС), которая содержит информацию, зафиксированную в результате предварительной проработки проекта. Это модель поверхности изделия, конструктивно-силовая схема, схема технологического членения, схемы трасс коммуникаций и др. Данные из БКС (и только из БКС) конструктор обязан использовать как исходные в своей работе. Кроме того, специалисты «ОКБ Сухого» отделили предварительную вспомогательную работу конструктора от ее результата. Вся работа по «электронной провязке» изделия делается в своей «рабочей» части и только потом модель, свободная от лишних построений, вносится в электронный макет разрабатываемого изделия. Этот подход обеспечивает возможность проведения контролируемых изменений на этапах взаимоувязки и согласования противоречивых требований.

Может сложиться впечатление, что PLM – дорогое решение, целесообразное и эффективное лишь для предприятий-гигантов, таких как «Сухой», Boeing или Toyota. На деле же принципы, лежащие в основе PLM, работают на предприятиях практически любого масштаба.

Есть уже сотни примеров удачного внедрения на базе других решений так называемого «среднего уровня», в том числе и от российских разработчиков, таких компаний, как «АСКОН», «Топ Системы», «ААЕМ», предоставляющих готовые PLM-комплексы. Весьма интересны также продукты компании Technologies, обеспечивающие построение PLM-среды на базе уже существующих разрозненных рабочих мест САПР от разных поставщиков.

Может показаться, что PLM – панацея, спасающая от ошибок в реальном изделии. В какой-то мере это действительно так, риск создать неудачный продукт при использовании PLM-технологий значительно снижается, но при одном очень важном условии. Это условие – компетентность специалистов, занятых созданием продукта. PLM не заменяет и не призван заменять специалистов, но значительно увеличивает эффективность их труда. Соответственно, имея в руках столь мощный инструмент, некомпетентный конструктор способен внести ошибку, которая, как снежный ком, вызовет цепочку других ошибочных или неоптимальных решений. Поэтому внедрение PLM – это не только закупка соответствующих программных систем, но и обязательная тщательная подготовка кадров, которые будут работать с ними.

Передаваемые параметры



Рис. 6.1. Интеграция данных САПР

Так как информационная модель PLM охватывает изделие и его жизнь целиком, то эта подготовка обязательна для всей иерархии участников создания продукта – от рядового техника до генерального директора. Естественно, нет необходимости каждому изучать абсолютно все элементы системы, но те, которые относятся к уровню его компетенции, а желательно и к смежным областям, просто необходимо.

6.1. Особенности процесса синтеза программных систем

Известно, что технологический цикл конструирования программной системы (ПС) включает три процесса – анализ, синтез и сопровождение.

В ходе анализа необходимо найти ответ на вопрос: «Что должна делать будущая система?». Именно на этой стадии закладывается фундамент успеха всего проекта. Известно множество неудачных реализаций из-за неполноты и неточностей в определении требований к системе.

В процессе синтеза формируется ответ на вопрос: «Каким образом система будет реализовывать предъявляемые к ней требования?». Выделяют три этапа синтеза: проектирование ПС, кодирование ПС, тестирование ПС (рис. 6.2).

Рассмотрим информационные потоки процесса синтеза.

Этап проектирования питают требования к ПС, представленные информационной, функциональной и поведенческой моделями анализа. Иными словами, модели анализа поставляют этапу проектирования исходные сведения для работы. Информационная модель описывает информацию, которую, по мнению заказчика, должна обрабатывать ПС. Функциональная модель определяет перечень функций обработки. Поведенческая модель фиксирует желаемую динамику системы (режимы ее работы). На выходе этапа проектирования – разработка данных, разработка архитектуры и процедурная разработка ПС.

Разработка данных – это результат преобразования информационной модели анализа в структуры данных, которые потребуются для реализации программной системы.

Разработка архитектуры выделяет основные структурные компоненты и фиксирует связи между ними.

Процедурная разработка описывает последовательность действий в структурных компонентах, т. е. определяет их содержание.

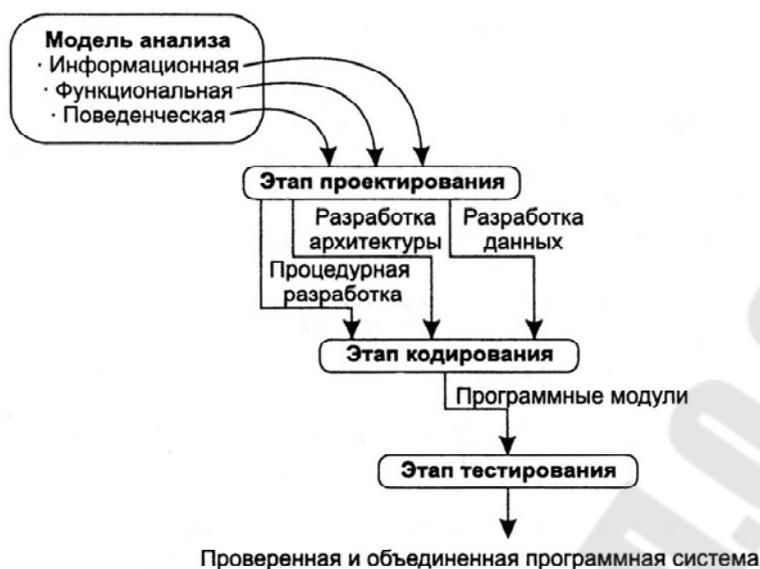


Рис. 6.2. Информационные потоки процесса синтеза программной системы

Далее создаются тексты программных модулей, проводится тестирование для объединения и проверки ПС. На проектирование, кодирование и тестирование приходится более 75 % стоимости конструирования ПС. Принятые здесь решения оказывают решающее воздействие на успех реализации ПС и легкость, с которой ПС будет сопровождаться.

Следует отметить, что решения, принимаемые в ходе проектирования, делают его стержневым этапом процесса синтеза. Важность проектирования можно определить одним словом – качество. Проектирование – этап, на котором «выращивается» качество разработки ПС. Справедлива следующая аксиома разработки: может быть плохая ПС при хорошем проектировании, но не может быть хорошей ПС при плохом проектировании. Проектирование обеспечивает нас такими представлениями ПС, качество которых можно оценить. Проектирование – единственный путь, обеспечивающий правильную трансляцию требований заказчика в конечный программный продукт.

Проектирование – итерационный процесс, при помощи которого требования к ПС транслируются в инженерные представления ПС. Вначале эти представления дают только концептуальную информацию (на высоком уровне абстракции), последующие уточнения приводят к формам, которые близки к текстам на языках программирования. Обычно в проектировании выделяют две ступени: предварительное проектирование и детальное проектирование. Предварительное проектирование формирует абстракции архитектурного уровня, детальное проектирова-

ние уточняет эти абстракции, добавляет подробности алгоритмического уровня. Кроме того, во многих случаях выделяют интерфейсное проектирование, цель которого – сформировать графический интерфейс пользователя (GUI). Схема информационных связей процесса проектирования приведена на рис. 6.3.



Рис. 6.3. Информационные связи процесса проектирования

Предварительное проектирование обеспечивает:

- идентификацию подсистем;
- определение основных принципов управления подсистемами, взаимодействия подсистем.

Предварительное проектирование включает три типа деятельности:

1. Структурирование системы – система структурируется на несколько подсистем, где под подсистемой понимается независимый программный компонент. Определяются взаимодействия подсистем.

2. Моделирование управления – определяется модель связей управления между частями системы.

3. Декомпозиция подсистем на модули – каждая подсистема разбивается на модули. Определяются типы модулей и межмодульные соединения.

Структурирование системы

Известны четыре модели системного структурирования:

- модель хранилища данных;
- модель клиент-сервер;
- трехуровневая модель;
- модель абстрактной машины.

В модели хранилища данных (рис. 6.4) подсистемы разделяют данные, находящиеся в общей памяти. Как правило, данные образуют БД. Предусматривается система управления этой базой.

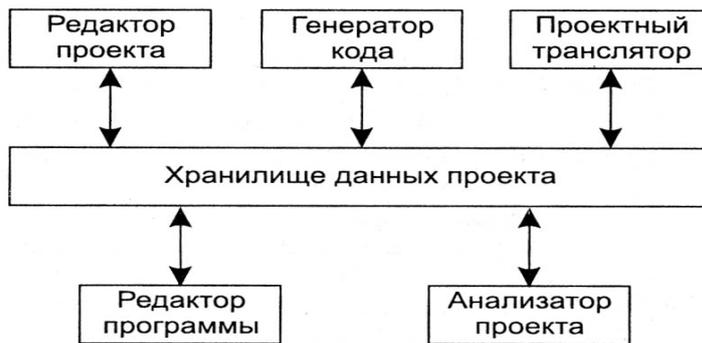


Рис. 6.4. Модель хранилища данных

Модель клиент-сервер используется для распределенных систем, где данные распределены по серверам (рис. 6.5). Для передачи данных применяют сетевой протокол, например TCP/IP.



Рис. 6.5. Модель клиент-сервер

Трехуровневая модель является развитием модели клиент-сервер (рис. 6.6).



Рис. 6.6. Трехуровневая модель

Уровень графического интерфейса пользователя запускается на машине клиента. Бизнес-логику образуют модули, осуществляющие функциональные обязанности системы. Этот уровень запускается на сервере приложения. Реляционная СУБД хранит данные, требуемого

уровня бизнес-логики. Этот уровень запускается на втором сервере – сервере базы данных.

Преимущества трехуровневой модели:

– упрощается такая модификация уровня, которая не влияет на другие уровни;

– отделение прикладных функций от функций управления БД упрощает оптимизацию всей системы.

Модель абстрактной машины отображает многослойную систему (рис. 6.7).

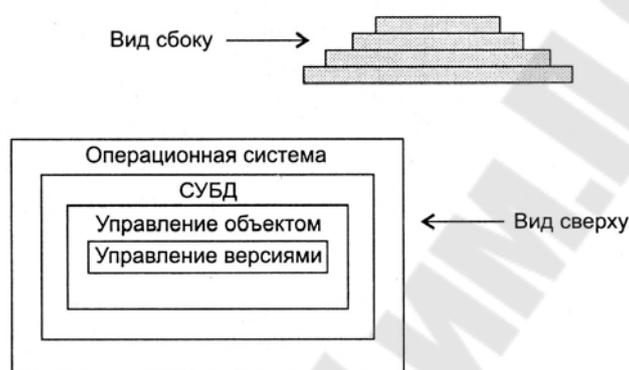


Рис. 6.7. Модель абстрактной машины

Каждый текущий слой реализуется с использованием средств, обеспечиваемых слоем-фундаментом.

Декомпозиция подсистем на модули

Известны два типа моделей модульной декомпозиции:

- модель потока данных;
- модель объектов.

В основе модели потока данных лежит разбиение по функциям.

Модель объектов основана на слабо сцепленных сущностях, имеющих собственные наборы данных, состояния и наборы операций. Очевидно, что выбор типа декомпозиции должен определяться сложностью разбиваемой подсистемы.

Модульность

Модуль – фрагмент программного текста, являющийся строительным блоком для физической структуры системы. Как правило, модуль состоит из интерфейсной части и части-реализации.

Модульность – свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей. Модульность – свойство ПО, обеспечивающее интеллектуальную возможность создания сколь угодно сложной программы.

7. МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ В СИСТЕМЕ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

7.1. Введение в метод конечных элементов

В реальных конструкциях почти всегда присутствуют сложные формы, состоящие к тому же из различных материалов. В качестве примера рассмотрим задачи, представленные на рис. 7.1. Рассчитать распределение напряжений в кронштейне (рис. 7.1, а) при помощи аналитических методов крайне сложно. Если же кронштейн изготовлен из композитного материала со сложными свойствами, задача становится практически неразрешимой. Непреодолимые затруднения возникают и при попытке вывести аналитическое выражение для распределения температур в объекте, изображенном на рис. 7.1, б.

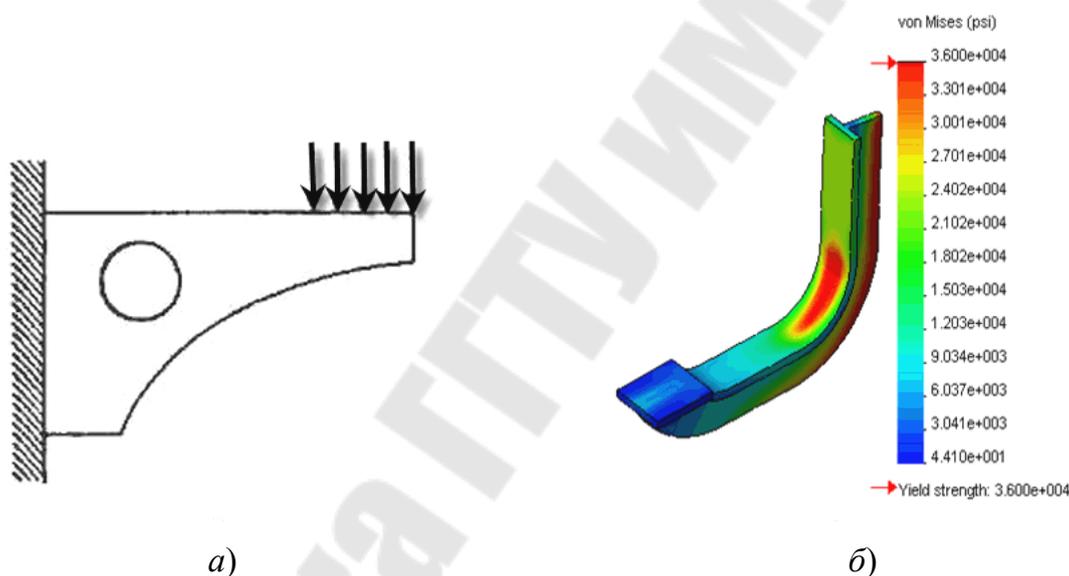


Рис. 7.1. Задачи, не имеющие аналитического решения

Метод конечных элементов (МКЭ), по всей видимости, является наиболее популярным численным методом решения таких задач. Универсальность этого метода удовлетворяет требованиям современных сложных систем конструирования, для которых обычно отсутствуют замкнутые решения уравнений равновесия. Анализ методом конечных элементов начинается с аппроксимации исследуемой области (области задачи) и деления ее на ячейки сетки. На рис. 7.2, а по углам каждой ячейки находятся узлы (черные точки). Такие ячейки называются конечными элементами. На рис. 7.2 представлены аппроксимации объектов с рис. 7.1 наборами конечных элементов (треугольных и четырехугольных).

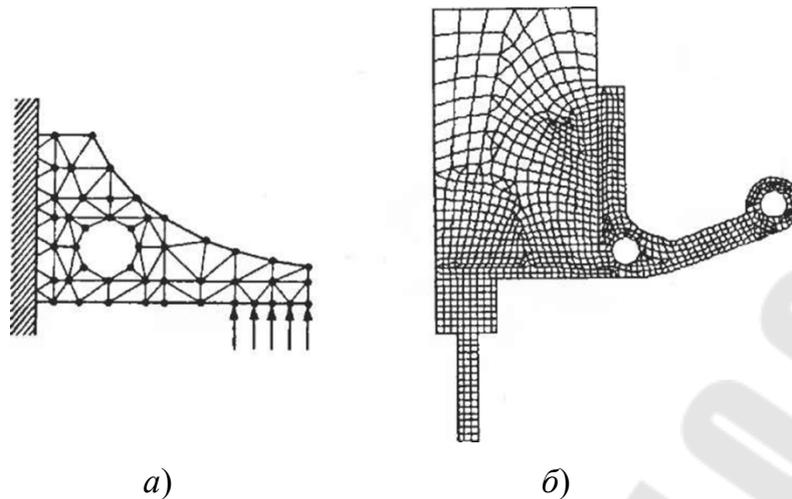


Рис. 7.2. Аппроксимация объектов конечными элементами

В этом примере аппроксимировали исходный объект треугольниками и четырехугольниками, однако, возможны и конечные элементы других типов. Выбор элементов определяется областью задачи, ее типом, а также конкретным пакетом анализа. Выбор подходящих элементов с нужным количеством узлов из библиотеки доступных элементов является одним из наиболее важных решений, которые приходится принимать пользователю пакета конечноэлементного анализа. Конструктору также приходится задавать полное количество элементов (другими словами, их размер). Общее правило состоит в том, что чем больше количество узлов и элементов (в h -версии) или чем выше степень функции формы (в p -версии), тем точнее оказывается решение, но тем дороже оно стоит с вычислительной точки зрения. Различные виды конечных элементов рассматриваются в разделе 8.3. Другая проблема – построение сетки, особенно для объекта сложной геометрии. Создание трехмерных сеток конечных элементов обычно представляет собой трудоемкий и кропотливый процесс. Сейчас ведутся активные разработки систем автоматизированного построения сеток, которые могли бы подключаться к системам геометрического моделирования. Такие системы позволили бы полностью интегрировать средства САМ и САЕ.

После аппроксимации исходного объекта конечными элементами с должным количеством узлов, каждому узлу сопоставляется неизвестная величина, которая ищется в процессе решения задачи. Например, на рис. 7.2, *a* неизвестными были бы смещения узлов по координатам x и y . Отсюда следует, что у каждого узла будет две степени свободы, а у задачи в целом будет $2n$ степеней свободы, если

число узлов равно n . Смещение в любой точке конечного элемента выводится из смещений его узлов при помощи функций формы, поэтому неизвестными могут быть только смещения узлов. Функции формы служат лишь для того, чтобы вычислять значения неизвестных внутри элемента по заданным значениям на его узлах. После вычисления смещений программа может перейти к расчету деформаций как частных производных от функции смещения, а по деформациям рассчитываются напряжения.

Аппроксимировав область задачи набором дискретных конечных элементов, необходимо задать характеристики материала и граничные условия для каждого элемента. Указав различные характеристики для разных элементов, можно анализировать поведение объекта, состоящего из разных материалов. Граничные условия (смещение, внешняя сила или температура) обычно задаются на внешней границе объекта. Эти условия должны быть выражены в виде значений смещения, силы или температуры в граничных узлах некоторых конечных элементов. После задания граничных условий для всех внешних узлов программа конечноэлементного анализа формирует систему уравнений, связывающую граничные условия с неизвестными (смещениями или температурой в узлах, или коэффициентами функции формы в p -версии), после чего решает эту систему относительно неизвестных.

После нахождения значений неизвестных пользователь получает возможность рассчитать значение любого параметра в любой точке любого конечного элемента по той же функции формы, которая использовалась при построении системы уравнений. Выходные данные программы анализа методом конечных элементов обычно представляются в числовой форме. В задачах механики твердых тел выходными данными являются смещения и напряжения. В задачах на теплоперенос выходными данными являются температуры и тепловые потоки через конкретные элементы. Однако по числовым данным пользователю бывает затруднительно получить общее представление о поведении соответствующих параметров. Графические изображения обычно более информативны, поскольку дают возможность изучить поведение параметров на всей области задачи. Анализ поведения параметров может производиться при помощи постпроцессора, который строит кривые и контурные графики переменных, по данным программы конечноэлементного анализа. Для задач строительной механики возможно отображение деформированных тел вместе с недеформированными. В этой области для систем автоматизированного

конструирования очень важными становятся функции компьютерной графики.

Многие конструкторы страдают чрезмерной верой в мощь этого метода, не имея представления о его ограничениях; они принимают неправильные результаты без тени сомнения. К преимуществам метода конечных элементов относится возможность работы с телами произвольной геометрии и неоднородными материалами. Однако суть метода состоит в делении области задачи на набор конечных элементов и поиске наилучшего решения, непрерывного «внутри» элементов, но имеющего возможность претерпевать скачки на их границах. Например, деформация на границе конечных элементов кронштейна (рис. 7.2, а) может испытывать скачок, невозможный с точки зрения физики. Величина такого скачка часто служит мерой точности решения, полученного методом конечных элементов. Неточности такого рода зависят от количества элементов, их размера и степени функции формы, используемой внутри каждого из элементов.

7.2. Решение задач проектирования с использованием программного пакета «ANSYS»

Генерация модели

Конечная цель конечноэлементного анализа состоит в математическом описании поведения реальной технической системы. Другими словами, такой анализ должен быть точной математической моделью некоторого материального прототипа. В широком смысле эта модель включает в себя все узлы, конечные элементы, свойства материала, реальные константы, граничные условия и другие особенности, которые используются для отображения физической системы.

В принятой для программы ANSYS терминологии под понятием *генерация модели* обычно подразумевается создание узлов и элементов, которые отображают пространственную конфигурацию и связи реальной системы. Таким образом, в нашем случае *генерация модели* означает *задание геометрической конфигурации узлов и элементов модели*. Программа ANSYS предлагает пользователю несколько различных методов построения модели, допуская при ее создании использование комбинации способов.

Последовательность шагов при построении модели

В общем случае процесс моделирования можно выполнять в соответствии с изложенной ниже основной схемой (более подробную

информацию, относящуюся к выделенным курсивом понятиям, можно найти в материале, изложенном далее).

1. Начиная с *планирования своего подхода* к решению задачи. Определите цели расчета; решите, какую основную форму будет иметь модель; выберите соответствующий тип элемента и необходимую плотность сетки. Такое планирование обычно выполняется до начала работы с программой ANSYS.

2. Построение модели начинается с обращения к препроцессору (PREP7). Наиболее часто при построении модели используются процедуры *твердотельного моделирования*.

3. Задайте *рабочую плоскость*.

4. При генерировании основных геометрических особенностей модели используйте *геометрические примитивы* и *булевы операции*.

5. Активизируйте соответствующую *систему координат*.

6. Задайте свойства твердотельной модели по принципу *снизу вверх*, т. е. сначала создайте необходимые *ключевые точки*, затем определите *линии, области* и *объемы* модели.

7. Для соединения отдельных областей модели в одно целое обращайтесь к *булевым операциям* или *управлению номерами*.

8. Создайте таблицы атрибутов элемента (*тип элемента, реальные константы, свойства материала* и *система координат элемента*).

9. Установите *указатели атрибутов элементов*.

10. Задайте параметры *управления качеством сетки*, если предполагается получить сетку желаемой плотности. Этот шаг требуется не всегда, т. к. при запуске программы размеры элемента задаются по умолчанию. (Если нужно, чтобы программа измельчала сетку автоматически, следует на этом шаге выйти из препроцессора и активизировать опцию *адаптивного построения сетки*).

11. Создайте узлы и элементы *построением сетки* для твердотельной модели.

12. После создания узлов и элементов вводятся такие особенности модели, как *контактные элементы типа поверхность-поверхность, связанные степени свободы* и *условия-ограничения*.

13. Сохраните относящиеся к модели данные в файле *Jobname.DB*.

14. Выйдите из препроцессора.

Твердотельное моделирование и прямая генерация

Для построения модели можно использовать два различных способа: *твердотельное моделирование* и *прямую генерацию*. При *твер-*

дотельном моделировании указываются геометрические границы модели, устанавливаются параметры управления размером и формой элементов, затем программе предписывается автоматическое построение всех узлов и элементов. При использовании способа *прямой генерации* созданию объектов модели предшествует задание местоположения каждого узла, а также размера, формы и связей элементов.

Хотя возможна некоторая автоматическая генерация задаваемых данных, метод прямой генерации является в основном «ручным», который требует отслеживания всех номеров узлов. Столь подробная «бухгалтерия» может стать утомительной при построении больших моделей, что делает возможным появление ошибок. Твёрдотельное моделирование, в общем случае, представляет собой более мощное и гибкое средство по сравнению с методом прямой генерации и, как правило, его использование для построения модели более предпочтительно.

Несмотря на многие преимущества твёрдотельного моделирования, при некоторых обстоятельствах может оказаться полезным применение метода прямой генерации. При построении модели можно использовать оба этих метода, легко переключаясь с одного на другой, более подходящий для данной области модели.

Чтобы помочь пользователю понять, какой метод более подходит в данной ситуации, приводится сравнение преимуществ и недостатков этих двух подходов.

Твёрдотельное моделирование (преимущества)

1. В основном подходит для больших или сложных моделей, особенно при построении пространственных моделей.

2. Позволяет работать с относительно малым числом данных.

3. Допускает геометрические операции (такие как перемещения и вращения), которые не могут быть выполнены с узлами и элементами.

4. Поддерживает использование «примитивных» поверхностей и объёмов (типа плоских многоугольных поверхностей и цилиндрических объёмов) и булевых операций (пересечение, вычитание и т. д.) для построения модели по принципу «снизу-вверх».

5. Обеспечивает возможность использования средств оптимизации программы ANSYS.

6. *Требуется* при адаптивном перестроении сетки.

7. *Требуется*, если нужно измельчить сетку после приложения к модели «твёрдотельных» нагрузок.

8. Позволяет быстро вносить изменения в геометрию модели.
9. Обеспечивает возможность менять расположение элементов (в этом случае пользователь не ограничен одним типом анализа).

Твердотельное моделирование (недостатки)

1. Иногда требует больших затрат времени работы центрального процессора.
2. Для малых и простых моделей может быть более громоздким, требуя больше данных ввода по сравнению с методом прямой генерации.
3. При некоторых обстоятельствах может оказаться несостоятельным (программа не сможет создать сетку конечных элементов).

Прямая генерация (преимущества)

1. Удобен для малых или простых моделей.
2. Предоставляет полный контроль над геометрией и нумерацией каждого узла и каждого элемента.

Прямая генерация (недостатки)

1. Требуется много времени едва ли не для самых простых моделей; объем данных, с которыми приходится работать, может оказаться обременительным.
2. Нельзя использовать при адаптивном перестроении сетки.
3. Делает процесс оптимизации разработки менее удобным.
4. Приводит к трудностям при необходимости модифицировать сетку (не могут быть использованы такие возможности программы, как рафинирование сетки, средства модуля SmartSizing и т. д.).
5. Может быть весьма утомительным, требуя уделять больше внимания каждой подробности сетки. Это может служить причиной появления ошибок.

Важность планирования

Приступив к созданию геометрической модели, пользователь принимает (сознательно или бессознательно) ряд решений, которые определяют, в какой степени его математическая модель отображает данную физическую систему, т. е. он отвечает на следующие вопросы: Каковы цели анализа? Будет ли создаваться модель всей системы или только ее части? Насколько подробной будет модель? Какой тип элементов будет использоваться? Насколько плотной должна быть сетка элементов? В основном, придется балансировать между затратами времени на этап вычислений (время работы центрального процессора и т. д.) и точностью результатов. Решения, принимаемые на стадии планирования, будут в значительной степени влиять на успех или неудачу затрачиваемых на анализ усилий.

Определение целей анализа

Этот первый шаг определяется не возможностями программы ANSYS, а образованием, опытом и профессиональным здравым смыслом пользователя. Только пользователь может определить, какими должны быть цели его анализа. Цели, заданные в самом начале анализа, оказывают влияние на последующие действия при генерации модели.

Выбор типа модели

Конечноэлементная модель может быть двумерной или трехмерной и состоять из точечных, линейных, поверхностных или объемных элементов. При необходимости можно использовать различные типы элементов, обращая внимание на совместимость соответствующих степеней свободы. Например, можно моделировать оболочечную конструкцию, усиленную ребрами жесткости, используя трехмерные оболочечные элементы для самой оболочки и трехмерные балочные элементы для ребер жесткости. Выбор размерности модели и типа элемента зачастую определяет наиболее предпочтительный метод генерации модели для данной задачи.

Линейные модели могут отображать двумерные, трехмерные балки или системы из труб так же, как двумерные модели – трехмерные осесимметричные оболочки. Твердотельное моделирование обычно не дает значительных преимуществ при построении линейных моделей; чаще всего они строятся с помощью методов прямой генерации.

Двумерные твердотельные модели используются для анализа тонкостенных плоских объектов (плоские напряжения), «бесконечно длинных» конструкций с постоянным поперечным сечением (плоская деформация) или осесимметричных массивных объектов. Хотя большинство двумерных моделей могут быть относительно просто созданы прямой генерацией, обычно удобнее пользоваться методом твердотельного моделирования.

Трехмерные оболочечные модели используются для тонкостенных пространственных структур. Хотя некоторые трехмерные оболочечные модели могут быть относительно просто созданы прямой генерацией, обычно удобнее пользоваться методом твердотельного моделирования.

Трехмерные твердотельные модели используются для толстостенных пространственных объектов, которые не имеют ни постоянного поперечного сечения, ни оси симметрии. Создание трехмерной

массивной модели методами прямой генерации обычно требует значительных усилий. Использование твердотельного моделирования почти всегда упрощает работу.

Линейные элементы и элементы высокого порядка. Библиотека элементов программы ANSYS включает два основных типа объемных и поверхностных элементов: *линейные* (с дополнительными формами или без) и *квадратичные*. Эти основные типы элементов схематично представлены на рис. 7.3.

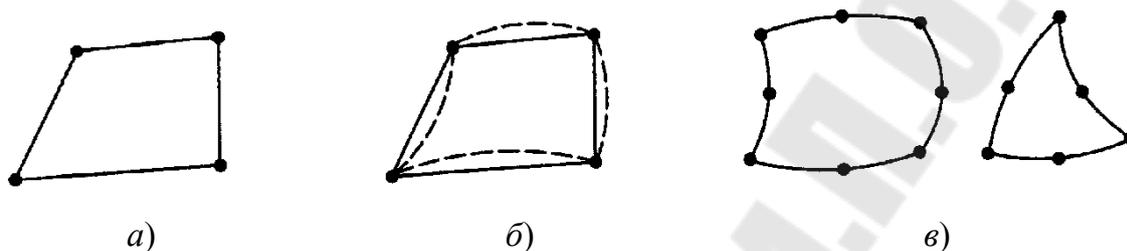


Рис. 7.3. Основные типы элементов программы ANSYS:
a – линейный изопараметрический; *б* – линейный изопараметрический с дополнительными формами; *в* – квадратичный

Линейные элементы (без срединных узлов). В прочностном анализе элементы с расположенными в углах узлами и с дополнительными функциями формы обычно приводят к точному решению при разумных затратах компьютерного времени. Важно избегать вырожденных форм этих элементов в критических областях, т. е. не допускать появления двумерных линейных элементов треугольной формы и трехмерных линейных элементов клинообразной или пирамидальной формы в областях высокого градиента расчетных величин. Также следует избегать появления линейных элементов с чрезмерными искажениями. При проведении *нелинейного* анализа лучшая точность решения при меньших затратах времени чаще достигается при использовании мелкой сетки линейных элементов по сравнению с крупной сеткой квадратичных элементов (рис. 7.4).

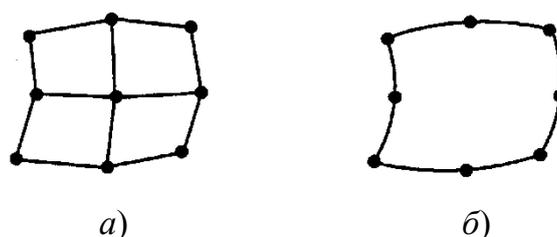


Рис. 7.4. «Сопоставимые» сетки линейных (*a*) элементов и квадратичных элементов (*б*)

При моделировании искривленных оболочек приходится выбирать между использованием искривленных (т. е. квадратичных) или плоских (т. е. линейных) оболочечных элементов. Каждый выбор имеет свои преимущества и недостатки. В большинстве случаев большинство задач может быть решено с высокой степенью точности за минимальное количество компьютерного времени при использовании плоских элементов. Однако нужно быть уверенным в том, что для моделирования соответствующей искривленной поверхности используется достаточное количество плоских элементов. Очевидно, чем меньше элемент, тем выше точность отображения геометрии. Рекомендуется, чтобы трехмерный плоский оболочечный элемент заменял собой дугу окружности не более 15° . Конические оболочечные (осесимметричные линейные) элементы должны быть ограничены дугой в 10° (или 5° , если находятся вблизи оси Y).

В большинстве других видов анализа (тепловой, магнитный и т. д.) линейные элементы работают почти так же хорошо, как элементы более высокого порядка, и требуют меньших затрат времени. При этом даже вырожденные элементы (треугольной и тетраэдрной форм) обычно приводят к точным результатам.

Квадратичные элементы (со срединными узлами). В линейном прочностном анализе при наличии вырожденных форм конечного элемента (т. е. треугольных двумерных элементов и клинообразных или тетраэдрных трехмерных элементов) использование квадратичных элементов обычно приводит к лучшим результатам при меньших затратах времени, чем использование линейных. Однако для правильного использования этих элементов необходимо знать их особенности.

Использование преимуществ симметрии. Многие объекты имеют симметрию некоторого рода; это может быть симметрия периодически повторяющейся структуры (например, равномерно расположенные ребра охлаждения на длинной трубе), симметрия отражения (литой пластиковый контейнер) или осесимметричность (электrolампа). Если тело симметрично во всех отношениях (геометрия, нагрузка, граничные условия и свойства материала), можно использовать это свойство для уменьшения размеров модели.

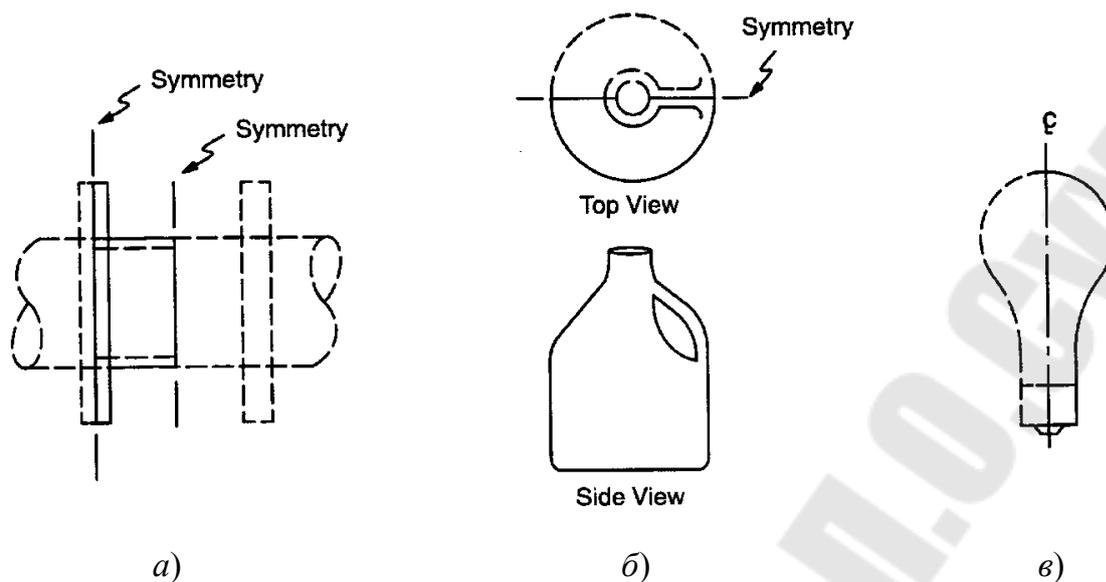


Рис. 7.5. Примеры симметрии:
a – симметрия повторения (охлаждающая труба); *б* – симметрия отражения (литой контейнер); *в* – осесимметричность (электролампа)

Замечания относительно осесимметричных конструкций. Любая конструкция, геометрия которой симметрична относительно центральной оси (оболочки или тела вращения), называется *осесимметричной*. Примерами могут служить прямые трубы, конусы, круглые пластины, купола и т. д.

Расчетные модели осесимметричных трехмерных конструкций могут быть представлены в эквивалентной двумерной форме. Можно ожидать, что анализ с использованием двумерной осесимметричной модели будет более точным, чем такой же анализ для трехмерной модели.

По определению, полностью осесимметричная модель испытывает действие только осесимметричных нагрузок. Однако во многих случаях осесимметричные конструкции могут подвергаться *неосесимметричным нагрузкам*. Следует использовать специальный тип элемента, известный как *осесимметричный гармонический*, чтобы создать двумерную осесимметричную модель с неосесимметричными нагрузками (Справочное руководство к программе *ANSYS Elements Reference*).

Особые требования к осесимметричным моделям

Ось симметрии должна совпадать с осью Y глобальной декартовой системы координат. Отрицательные узловые координаты X не разрешаются. В глобальной декартовой системе координат ось Y совпадает с направлением оси вращения, ось X – с радиальным направлением, а ось Z соответствует кольцевому направлению. Модель должна быть создана из соответствующих типов элементов:

1. Для осесимметричных моделей используются подходящие объемные с опцией KEYOPT(3) = 1 и/или осесимметричные оболочечные элементы. Кроме того, в такую модель могут быть включены элементы связи, контактные, комбинированные и поверхностные элементы (для программы эти «другие» элементы становятся также осесимметричными, если в модели присутствуют осесимметричные объемные или оболочечные элементы). Если отдельные типы элементов Справочного руководства *ANSYS Elements Reference* не рассматриваются как осесимметричные, эти элементы в осесимметричном анализе не следует использовать.

2. Для осесимметричных гармонических моделей используются *только* осесимметричные гармонические элементы.

3. Для моделей, содержащих двумерные твердотельные элементы, следует использовать, по крайней мере, два элемента по толщине, если нужно учесть сдвиговые эффекты.

Некоторые рекомендации и ограничения

Если рассматриваемая деталь имеет отверстие по оси симметрии, следует предусмотреть надлежащее расстояние между осью Y и двумерной осесимметричной моделью.

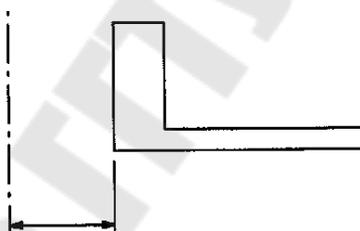


Рис. 7.6. Смещение по оси X соответствует радиусу отверстия

Подробности описания модели. Незначительные детали геометрии не должны учитываться при построении твердотельной модели, т. к. они без необходимости будут только усложнять задачу. Однако в некоторых случаях такие «мелкие» подробности, как галтели или отверстия могут стать областью максимальных напряжений и оказаться важными – в зависимости от целей анализа. Следует адекватно представлять себе предполагаемую реакцию конструкции, чтобы компетентно принять решение о включении тех или иных ее подробностей в расчетную модель.

В некоторых случаях симметрия конструкции нарушается только несколькими мелкими деталями. Для получения симметричной модели можно игнорировать их (или, наоборот, считать симметрич-

ными). При этом следует сопоставлять преимущества за счет упрощения модели с потерями в точности решения.

Выбор плотности сетки. При выполнении конечно-элементного анализа часто возникает вопрос о том, насколько мелкой должна быть сетка элементов, чтобы получить приемлемые результаты? К сожалению, никто не сможет дать определенный ответ на этот вопрос. Некоторые практические приемы, которые можно использовать для его решения, состоят в следующем.

1. Для построения сетки используйте опцию *адаптивного измельчения*, которая использует энергетический критерий допустимой оценки погрешности. (Этот прием доступен только для линейных статических или стационарных тепловых задач. «Допустимый» уровень погрешности зависит от требований, предъявляемых к анализу). Для использования адаптивного перестроения сетки *требуется* твердотельное моделирование.

2. Сравнивайте результаты предварительного анализа с независимо полученными экспериментальными или известными аналитическими решениями. Измельчайте сетку в тех областях, где расхождение между известными и расчетными результатами слишком велико. (Сетку на поверхности модели можно локально измельчать с помощью команд **NREF**, **EREF**, **KREF** и **LREF** или используя маршрут меню **Main Menu>Preprocessor>-Meshing-Refine>тип объекта**).

3. Выполняйте предварительный анализ, используя сетку «разумных» размеров. Повторите анализ, увеличив вдвое число элементов в критических областях, и сравните оба решения. Если получены почти одинаковые результаты, то сетку, скорее всего, можно считать адекватной. Если результаты существенно различаются, то, возможно, требуется дальнейшее измельчение сетки. Следует увеличивать число элементов, пока не будут получены почти идентичные результаты.

4. Если в процессе измельчения сетки сопоставление результатов показывает, что более мелкая сетка требуется только для части модели, следует использовать *метод подмоделей*, чтобы «увеличить» размеры критических областей.

Плотность сетки является чрезвычайно важным элементом расчета. Если сетка слишком крупная, результаты могут содержать грубые ошибки. При слишком мелкой сетке будут напрасно расходоваться компьютерные ресурсы, приводя к чрезмерным затратам времени, а модель может оказаться слишком громоздкой для используемой

компьютерной системы. Чтобы избежать подобных проблем, всегда следует оценивать плотность сетки перед генерацией модели.

Сетка для твердотельной модели. Процедура создания узлов и конечных элементов сетки состоит из трех основных этапов:

- задание атрибутов для элементов сетки;
- установка средств управления качеством сетки (по выбору);
- генерирование сетки.

Второй этап используется не всегда, поскольку в большинстве случаев управление качеством сетки используется по умолчанию. Если такой контроль не задается явно, то программа использует задаваемые по умолчанию установки команды **DESIZE** для создания так называемой произвольной сетки. В качестве альтернативы можно использовать программное средство разумного выбора размеров сетки, так называемую процедуру SmartSize.

Произвольная и упорядоченная сетки. Перед построением сетки и даже до создания геометрической модели имеет смысл подумать о том, какая сетка элементов – произвольная или упорядоченная – пригодна для данного вида анализа. *Произвольная сетка* представляет собой сетку без ограничений на форму элементов и без специальной структуры их расположения. *Упорядоченная сетка*, в отличие от произвольной, накладывает ограничения на форму элементов и структуру их размещения. Такая сетка состоит только из четырехугольных (плоских) или только шестигранных (объемных) элементов. Кроме того, упорядоченная сетка имеет, как правило, регулярную структуру с явными рядами элементов. Если желательно получить именно такой тип сетки, то пользователь должен строить геометрическую модель как совокупность достаточно правильных объемов и/или поверхностей, для которых приемлема упорядоченная сетка элементов.

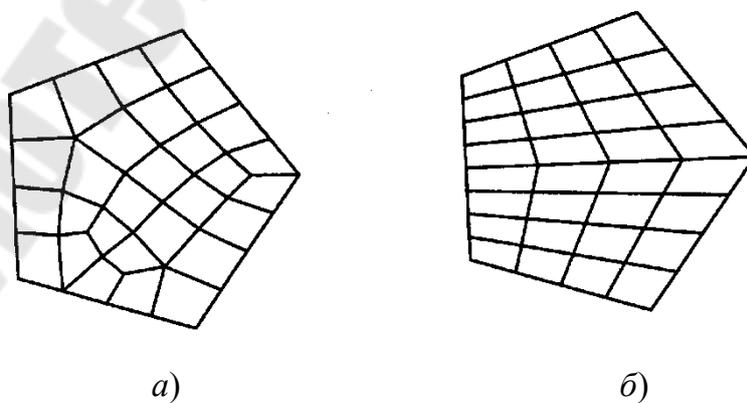


Рис. 7.7. Произвольная (а) и упорядоченная (б) сетки

Задание атрибутов для элементов. Перед генерацией сетки узлов и конечных элементов следует задать соответствующие *атрибуты элементов*. Это означает, что нужно задать следующее:

- тип элемента (т. е. BEAM3, SHELL61 и т. п.);
- набор реальных констант (обычно относящихся к геометрии элемента: толщина, площадь поперечного сечения и др.);
- свойства материала (модуль упругости, теплопроводность и т. д.);
- систему координат элемента.

Таблицы атрибутов. Чтобы присвоить атрибуты элементам, требуется создать таблицы атрибутов. К таким таблицам обычно относятся перечни типов конечных элементов (команда **ET** или маршрут меню **Main Menu>Preprocessor>Element Type>Add/Edit/Delete**), значений реальных констант (команда **R** или маршрут меню **Main Menu>Preprocessor>RealConst**) и свойств материала (команда **MP** или маршрут меню **Main Menu>Preprocessor>MaterialProps>опция**).

Также может быть создана таблица координатных систем с помощью таких команд, как **LOCAL**, **CLOCAL** и т. д. (маршрут меню **Utility Menu>WorkPlane>Local Coordinate Systems>Create Local CS>опция**). Эта таблица может быть использована для «привязки» системы координат к элементам. (Подобным образом можно связать координатную систему не с каждым типом элемента, см. описание элементов в пособии *ANSYS Elements Reference*).

Таблицы атрибутов можно представить в виде, показанном на рис. 7.8. Более подробная информация, касающаяся создания таблиц атрибутов, содержится в руководстве по основным процедурам анализа *ANSYS Basic Analysis Procedures Guide*.

Type	Real	Mat	ESYS
STIF3	1 A1, I1, H1	1 EX1, ALPX1, ...	0 Global Cartesian
	2 A2, I2, H2	2 EX2, ALPX2, ...	1 Global Cyлинд.
	3 A3, I3, H3	3	2 Global Sphere
•	•	•	1 (Local)
•	•	•	1
•	•	•	2
			••
	n	p	q

Рис. 7.8. Таблицы атрибутов конечных элементов:
числа n, p, q – номера ссылок

Содержимое таблиц можно просмотреть, используя команды **ETLIST** (таблица типов), **RLIST** (таблица констант), **MPLIST** (таблица свойств) или выбрав эквивалентный маршрут меню **Utility Menu>List>Properties>тип свойства**. К таблице координатных систем можно обратиться с помощью команды **CSLIST** (маршрут **Utility Menu>List>Other>Local Coord Sys**).

Присвоение атрибутов перед построением сетки. После создания таблиц атрибутов пользователь может присвоить атрибуты конечных элементов различным частям своей геометрической модели, используя в качестве указателей соответствующие номера ссылок этих таблиц. В результате такими указателями оказываются номера материалов (**MAT**), номера наборов реальных констант (**REAL**), номера типов конечных элементов (**TYPE**) и номера координатных систем (**ESYS**). Атрибуты можно присвоить выбранным объектам твердотельной модели или задать набор атрибутов по умолчанию (последний будет использоваться для конечных элементов, создаваемых при последующих операциях построения сетки).

Присвоение атрибутов объектам твердотельной модели. Возможность присвоения атрибутов объектам твердотельной модели позволяет предварительно задавать атрибуты для каждой области модели. Этим самым можно избежать необходимости переустанавливать атрибуты в процессе построения сетки. (Очистка твердотельных объектов от узлов и конечных элементов не отменяет атрибутов, непосредственно присвоенных этим объектам).

Для прямого присвоения атрибутов объектам твердотельной модели используется один из следующих способов.

1. Присвоение атрибутов ключевым точкам.

Команда: **KATT**

Маршрут: **Main Menu>Preprocessor>-Meshing-Attributes>All**

Keypoints;

Main Menu>Preprocessor>-Meshing-Attributes>Picked KPs

2. Присвоение атрибутов линиям.

Команда: **LATT**

Маршрут: **Main Menu>Preprocessor>-Meshing-Attributes>All**

Lines;

Main Menu>Preprocessor>-Meshing-Attributes>Picked Lines

3. Присвоение атрибутов поверхностям.

Команда: **AATT**

Маршрут: **Main Menu>Preprocessor>-Meshing-Attributes>All Areas;**

Main Menu>Preprocessor>-Meshing-Attributes>Picked Areas

4. Присвоение атрибутов объемам.

Команда: **VATT**

Маршрут: **Main Menu>Preprocessor>-Meshing-Attributes>All Volumes;**

Main Menu>Preprocessor>-Meshing-Attributes>Picked Volumes

5. Присвоение атрибутов по умолчанию.

Можно присвоить атрибуты по умолчанию, обратившись к нужным записям таблицы атрибутов. Те указатели, которые действуют во время создания конечных элементов модели (т. е. при инициировании процедуры построения сетки), используются программой для присвоения атрибутов из таблиц объектам твердотельной модели и данным конечным элементам. Атрибуты, присвоенные непосредственно объектам твердотельной модели (как описано выше), подавляют атрибуты по умолчанию. Если объекты твердотельной модели очистить от узлов и элементов, то атрибуты по умолчанию сохраняются.

Для присвоения атрибутов по умолчанию используется один из следующих способов:

Команды: **TYPE, REAL, MAT** и **ESYS**

Маршруты: **Main Menu>Preprocessor>-Meshing-Attributes>Default Attribs;**

Main Menu>Preprocessor>-Modeling>Create>Elements>Elem Attributes;

Main Menu>Preprocessor>-Modeling-Operate>Extrude/Sweep>Default Attribs

Управление построением сетки. Средства управления построением сетки, используемые программой ANSYS по умолчанию, могут дать сетку, вполне адекватную для анализируемой модели. В этом случае нет необходимости задавать такие средства явным образом. Однако если такие средства предполагается использовать, их следует определить до построения сетки.

Средства управления дают возможность устанавливать форму конечных элементов, расположение срединных узлов и размеров элементов при построении сетки конечных элементов на твердотельной модели. Этот этап является одним из наиболее важных для всего выполняемого анализа, поскольку принимаемые на этой стадии решения могут существенным образом сказаться на точности результатов и затратах на проведение расчета.

Форма конечных элементов. Как минимум следует задать перечень допустимых форм конечных элементов, если планируется строить сетку из элементов, которые могут иметь более одной формы. Например, многие поверхностные элементы могут быть как треугольными, так и четырехугольными в пределах одной и той же области модели. Объемные элементы часто бывают шестигранными («кирпич») или четырехгранными, но в одной модели одновременно использовать обе формы не рекомендуется. Для указания форм элемента используется один из следующих способов:

Команда: **ESHAPE**

Маршрут: **Main Menu>Preprocessor>-Meshing-Shape & Size>Element Shape**

Команда **ESHAPE** также контролирует размещение срединных узлов при использовании квадратных элементов.

В некоторых случаях для создания сеточной модели оказывается достаточно команды **ESHAPE** и соответствующих команд построения сетки (**LMESH**, **AMESH**, **VMESH** или **Main Menu>Preprocessor > Meshing-Mesh>опция сетки**). Размер каждого элемента будет определяться по умолчанию спецификациями команд **SMRTSIZE** или **DESIZE**. Так, например, для показанной на рис. 7.9 слева твердотельной модели одной командой **VMESH**, создается сеточная модель, приведенная справа. Для приведенной выше модели выбираемые программой размеры элементов не обязательно окажутся адекватными для данного вида анализа. Одним из способов перестроения сетки является изменение размера элемента, установленного по умолчанию в команде **SMRTSIZE**, и повторное генерирование сетки.

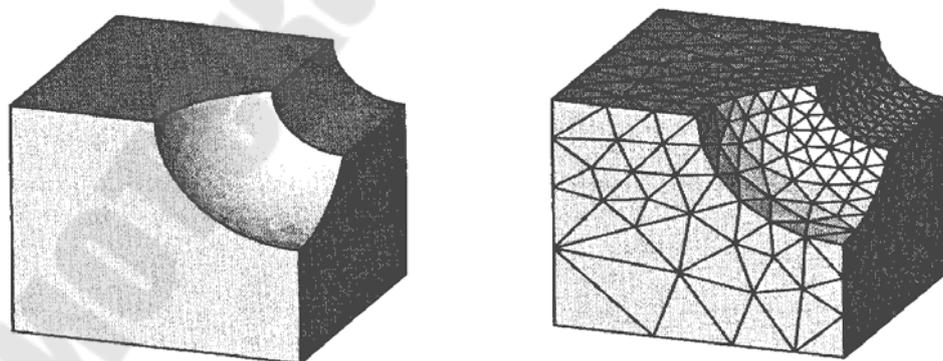


Рис. 7.9. Сетка из элементов по умолчанию

Размер элемента для произвольной сетки. С помощью программного средства разумного выбора размеров конечных элементов (**SmartSizing**) выбираются исходные значения для операций построения

ния произвольной сетки. Это средство дает сеточному генератору наилучшие возможности построить элементы приемлемых размеров. Управляемое командой **SMRTSIZE**, оно обеспечивает построение как грубой, так и мелкой сетки для элементов h - и p -методов.

При построении произвольной сетки по умолчанию используется команда **DESIZE**. Однако в таких случаях целесообразно обратиться к процедуре SmartSizing, для чего достаточно указать уровень размеров элемента в команде **SMRTSIZE** (см. ниже).

Преимущества процедуры SmartSizing. Алгоритмом этой процедуры предусмотрено определение прежде всего размеров ребер элементов для всех граничных линий модели, т. е. линий, ограничивающих те поверхности и объемы модели, для которых будет строиться сетка. Затем эти размеры уточняются для моделирования закруглений и других подробностей геометрии. Поскольку размеры всех объектов определяются до нанесения сетки, ее качество не зависит от порядка, в котором она строится для отдельных поверхностей и объемов модели. (Для получения сетки наилучшего качества ее следует строить сразу для всей геометрической модели).

Если для создания сетки на поверхности используются четырехсторонние элементы, процедура SmartSizing пытается выбрать равное число разбиений на границах этой поверхности для того, чтобы обеспечить построение сетки из четырехугольников. Треугольные элементы будут присутствовать в такой сетке только вынужденно.

Средства контроля, используемые процедурой SmartSizing. Процедура выбора размеров конечных элементов располагает двумя категориями средств контроля качества сетки: основной и расширенной.

Основная категория. Чтобы использовать возможности этой категории, достаточно задать признак размера сетки от **1** (мелкая) до **10** (крупная). Программа автоматически устанавливает ряд управляющих параметров, обеспечивающих построение сетки требуемого размера. Для задания размера сетки используется один из следующих способов:

Команда: **SMRTSIZE, SIZLVL**

Маршрут: **Main Menu>Preprocessor>-Meshing-Mesh> SmartSize Levels**

На рис. 7.10 показаны конечноэлементные модели при различных установках размера сетки, включая значение **6** по умолчанию. Расширенная категория. Пользователь может предпочесть эту категорию средств контроля, требующую ручной установки управляющих параметров. Это дает возможность «причесать» сетку в соответствии

с потребностями. Можно устранить наличие слишком мелких ячеек сетки и малых углов между сторонами элемента с помощью управляющих ключей, а также осуществить укрупнение и модификацию сетки. Кроме того, можно задать начальный размер элемента для процедуры SmartSizing с помощью команды ESIZE.

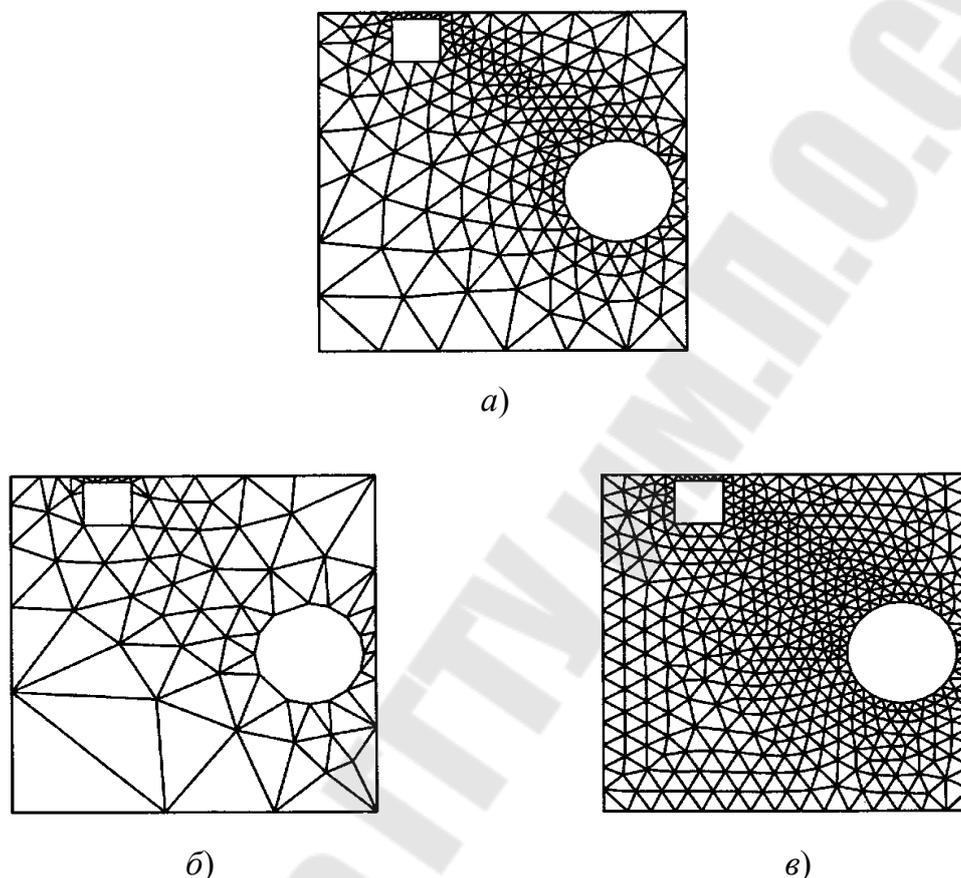


Рис. 7.10. Сетка при различных значениях параметра *SIZLVL*:
а – размер по умолчанию = 6; б – размер = 8 (грубая);
в – размер = 4 (точная)

Для обращения к расширенной категории средств управления используется один из следующих способов:

Команды: **SMRTSIZE** и **ESIZE**

Маршрут: **Main Menu>Preprocessor>-Meshing-Shape & Size>-SmartSize-Adv Opts**

Пересечение с другими средствами контроля. Локальные средства контроля, рассматриваемые в следующем пункте раздела, могут использоваться в сочетании с возможностями процедуры SmartSizing. В том случае, если установлены конфликтующие размеры элементов, алгоритмом процедуры предусмотрены следующие действия.

Используется размер элемента, установленный по длине стороны (команда **LESIZE** или маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-Line-опция**).

Предписывается размер элемента, установленный по ключевым точкам (команда **KESIZE** или маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-Keypoints-опция**), но он может быть изменен для отображения скруглений или других мелких подробностей геометрии.

Если установлен глобальный размер элемента (команда **ESIZE** или маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-Global-Size**), то он будет изменен для отображения скруглений или других мелких подробностей геометрии. Если желательно иметь непротиворечивые размеры, следует использовать глобальный способ задания и отключить процедуру SmartSizing (команда **SMRTSIZE**, **OFF** или маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-SmartSize-Basic**).

Размеры, установленные по умолчанию командой **DESIZE** (маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-Global-Map Mesh Def**), игнорируются при включении процедуры SmartSizing.

Размеры по умолчанию для упорядоченной сетки. Команда **DESIZE** позволяет изменить следующие установки, задаваемые по умолчанию: минимальное и максимальное число элементов, добавляемых к линии модели, не разделенной на элементы; максимальный угол между сторонами элемента; минимальная и максимальная длина ребра. Команда **DESIZE** (маршрут **Main Menu>Preprocessor>-Meshing-Shape & Size>-Global-Other**) всегда используется для контроля размеров элементов при построении упорядоченной сетки. Установки этой команды используются также по умолчанию для произвольной сетки. Рекомендуется, однако, вместо операций построения произвольной сетки использовать процедуру SmartSizing [**SMRTSIZE**].

В качестве примера на рис. 7.11, *а* представлена упорядоченная сетка, построенная с использованием настроек по умолчанию, сохраненными без изменения при входе в программу. Сетка на рис. 7.11, *б* создана путем модификации значений минимального числа элементов (**MINL**) и максимума угла между сторонами (**ANGL**) в команде **DESIZE**.

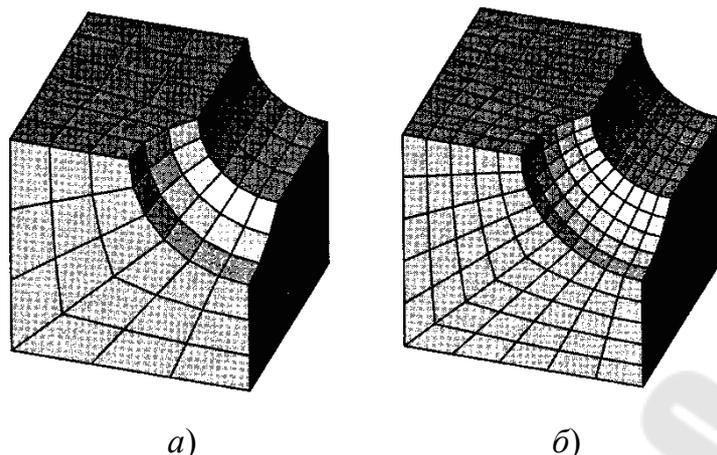


Рис. 7.11. Влияние изменений установок по умолчанию

Для больших моделей разумно обращаться к результатам использования команды **DESIZE**. Это можно сделать, вызвав на экран графическую информацию, касающуюся разбивки контурных линий модели (рис. 7.12). Выполняются следующие шаги:

1. Строится твердотельная модель.
2. Задается тип конечного элемента.
3. Задаются допускаемые формы элемента.
4. Вводится команда **LESIZE,ALL** (используется разбиение линий модели в соответствии со спецификациями команды **DESIZE**).
5. Запрашивается график линий [**LPLOT**].

Например:

```
ET,1,45          ! Восьмиузловой четырехсторонний элемент
ESHAPE,2       ! Разрешена только форма "кирпич"
LESIZE,ALL     ! Установки команды DESIZE
LPLOT
```

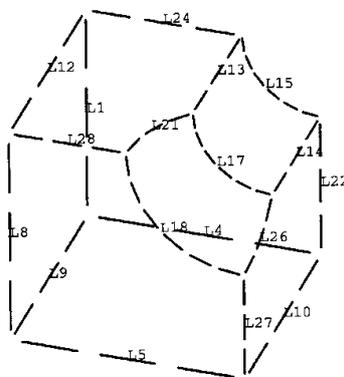


Рис. 7.12. Результат использования установок по умолчанию

Если полученная сетка представляется слишком грубой, ее можно изменить модификацией установок по умолчанию (рис. 7.13):

DESIZE,5,,30,15 Изменение размеров, установленных по умолчанию

LESIZE,ALL,,,,,1 ! Принудительное деление линий

LSPLOT

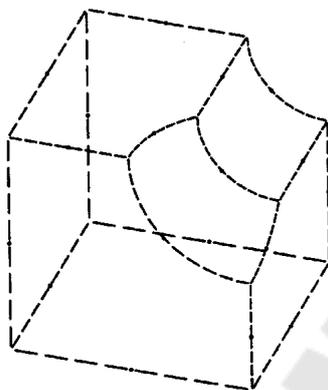


Рис. 7.13. Результат модифицирования

Локальные средства построения сетки. Во многих случаях сетка, созданная с применением задаваемых по умолчанию параметров, не соответствует физической сути проводимого анализа. Примерами могут служить сингулярные задачи или задачи расчета концентрации напряжений. При решении такого рода проблем пользователь вынужден уделять построению сетки более пристальное внимание. В его распоряжении имеются следующие средства управления размерами конечных элементов.

- Глобальные размеры элемента в виде длины ребра элемента, которые задаются на линиях, ограничивающих поверхности модели, или определяются числом деления этих линий.

Команда: **ESIZE**

Маршруты: **Main Menu>Preprocessor>-Meshing-Shape & Size>-Global-Size;**

MainMenu>Preprocessor>-Modeling-Operate>Extrude/Sweep>Size

- Размеры элемента вблизи указанных ключевых точек.

Команда: **KESIZE**

Маршруты: **Main Menu>Preprocessor>-Meshing-Shape & Size>-Keypoints-All KPs;**

MainMenu>Preprocessor>-Meshing-Shape & Size>-Keypoints-Cir Size

- Число элементов на указанных линиях.

Команда: **LESIZE**

Маршруты: **Main Menu>Preprocessor>-Meshing-Shape & Size>-Lines-All Lines;**

Main Menu>Preprocessor>-Meshing-Shape & Size>-Lines-Picked KLines;

Main Menu>Preprocessor>-Meshing-Shape & Size>-Lines-Cir Size

Все эти спецификации размеров можно использовать вместе. В случае конфликта команд действуют иерархические соглашения. Приоритеты в некоторой степени зависят от того, в соответствии с какой командой, **DESIZE** или **SMRTSIZE**, устанавливаются размеры по умолчанию.

*Иерархия при использовании команды **DESIZE**.* Вдоль любой контурной линии модели размеры элементов определяются следующим образом:

– предпочтение отдается делению контурных линий модели, заданному командой **LESIZE**;

– если деление линий отсутствует, то используется спецификация команды **KESIZE** для ключевых точек (если они заданы);

– если размер не задан делением линии или указанием ключевых точек, то используется спецификация команды **ESIZE**;

– если не используется ни одна из перечисленных возможностей, то размеры элементов контролируются установками команды **DESIZE**.

*Иерархия при использовании команды **SMRTSIZE**.* Вдоль любой контурной линии модели размеры элементов определяются следующим образом:

– предпочтение отдается делению контурных линий модели, заданному командой **LESIZE**;

– если деление линий отсутствует, то используется спецификация команды **KESIZE** для ключевых точек (если они заданы), но размеры элемента могут быть изменены для моделирования закруглений и других мелких подробностей геометрии;

– если размер элемента не задан делением линии или указанием ключевых точек, то для выбора его начальной величины используется спецификация команды **ESIZE**, но размер может быть изменен для моделирования закруглений и других мелких подробностей геометрии;

– если не используется ни одна из перечисленных возможностей, то размеры элементов контролируются установками команды **SMRTSIZE**.

Для четырехгранных элементов, в дополнение ко всем доступным локальным средствам управления качеством сетки, можно использовать двумерное рафинирование (облагораживание). Достаточно на поверхности нужного объема построить сетку из треугольных элементов, измельчить ее в нужных местах, а затем сгенерировать объемную сетку. Эти объемные конечные элементы будут соответствовать сетке на поверхности модели. Разумеется, перед получением решения следует удалить поверхностные элементы.

Если выполняется линейный прочностной или линейный стационарный тепловой анализы, то можно позволить программе самой управлять размером сетки конечных элементов и адаптировать ее таким образом, чтобы получить погрешность решения меньше заданной величины.

Внутренние средства построения сетки. Рассмотренные до сих пор средства построения сетки ограничивались заданием размера элемента на контурах твердотельной модели команды (**LESIZE**, **ESIZE** и др.). Вместе с тем имеется возможность управлять процессом создания сетки внутри областей модели, не имеющих линий, с помощью которых задается размер элемента. Для этого можно использовать один из следующих способов:

Команда: **МОРТ**

Маршрут: **Main Menu>Preprocessor>-Meshing-Shape & Size>-Global-Area Cntris**

Для перехода от мелкой сетки на границах модели к более крупной во внутренних ее областях используйте опцию *Lab=EXPAND* команды **МОРТ** (рис. 7.14).

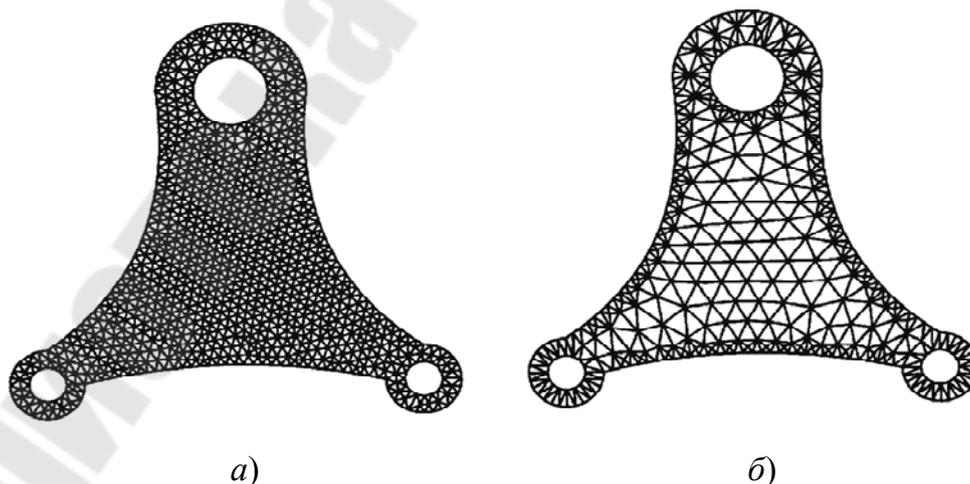


Рис. 7.14. Конечноэлементная модель без укрупнения и с укрупнением сетки:

a – без укрупнения сетки (1782 элементов);

б – результат **МОРТ,EXPAND,2.5** (614 элементов)

Конечноэлементная сетка на рис. 7.14, *а* построена целиком с помощью команды **ESIZE** (маршрут **Main Menu>Preprocessor>Meshing-Shape & Size>-Global-Size**) и имеет хорошую структуру, но для заполнения всей области потребовалось 1782 элемента. (Данная модель состоит из одной поверхности). Модель на рис. 7.14, *б* построена с помощью опции укрупнения сетки (*Lab=EXPAND*) в команде **МОПТ**; меньшее число элементов получено за счет увеличения размера сетки от мелкой на границах модели до более крупной во внутренней области. Однако некоторые элементы этой сетки имеют неудовлетворительное отношение сторон (например, элементы вокруг отверстий малого радиуса). Еще одним недостатком сетки, представленной на рис. 7.14, *б*, является то, что переход от мелких элементов к крупным происходит слишком быстро.

Для улучшения сетки на рис. 7.14, *б* требуется выполнить постепенный переход от элементов малого размера к крупным. Опция *Lab=EXPAND* команды **МОПТ** может использоваться для управления процессом изменения размера сетки. На рис. 7.15 показана сетка для этой же модели, полученная в результате применения команды **МОПТ, TRANS,1.3** в дополнение к установкам команды **МОПТ**, которые использовались на предыдущем этапе. Эта сетка, по сравнению с приведенной на рис. 7.15, *а*, имеет намного меньше элементов, а изменение размеров ячеек происходит достаточно плавно. Кроме того, отношение сторон конечных элементов значительно лучше, чем для сетки на рис. 7.15, *б*.



Рис. 7.15. Использование укрупнения и переходов

8. БЫСТРОЕ ПРОТОТИПИРОВАНИЕ И ИЗГОТОВЛЕНИЕ

Хотя геометрическая модель использовалась как источник общих данных, провесы проектирования и производства не были непосредственно интегрированы в процедуру обработки. Иными словами, чтобы станок с ЧПУ мог выполнить обработку по данным геометрической модели, требовался ряд промежуточных шагов: планирование процессов, рассмотрение конструкций зажимов и креплений, выбор оборудования.

Еще один способ использования геометрической модели в производстве – это быстрое прототипирование. Существуют разные процессы быстрого прототипирования, но все их объединяет то, что прототип изготавливается путем послойного наложения композитного материала. Основное преимущество быстрого прототипирования состоит в том, что прототип создается за один прием, исходными данными для него служит непосредственно геометрическая модель детали. Таким образом отпадает необходимость в планировании последовательности технологических процессов, специальном оборудовании для обработки материалов, транспортировке от станка к станку и т. д. Однако по сравнению с обработкой на станке с ЧПУ этот процесс имеет существенный недостаток – ограниченность выбора материалов. Поскольку станки с ЧПУ способны обрабатывать большую часть доступных промышленных материалов, включая металлы, то физические объекты, изготовленные методом быстрого прототипирования, используются главным образом в качестве прототипов или шаблонов для других производственных процессов.

8.1. Обзор

После появления систем твердотельного моделирования в начале 70-х гг. XX в. делались попытки генерировать физические объекты непосредственно из геометрических данных, не прибегая к использованию традиционных инструментов. Новая технология получила название быстрое прототипирование (rapid prototyping), а также послойное изготовление (layered manufacturing), трехмерная печать (3D printing), настольное изготовление (desktop manufacturing) и изготовление объемных деталей произвольной формы (solid freeform manufacturing). С того времени эта технология шагнула далеко вперед, найдя множество применений на производстве помимо изготовления прототипов. Исходя из этого, более удачным обозначением для нее в настоящий момент пред-

ставляется термин быстрое прототипирование и изготовление (rapid prototyping and manufacturing), или БПИ, который мы и будем использовать в этой главе.

В основе своей процессы быстрого прототипирования и изготовления состоят из трех шагов: формирование поперечных сечений изготавливаемого объекта, послойное наложение этих сечений и комбинирование слоев. Таким образом, чтобы создать физический объект, этим процессам требуются данные лишь о поперечных сечениях; кроме того, исчезают следующие проблемы, часто возникающие в связи с другими производственными процессами.

Отпадает необходимость в топологическом проектировании и распознавании по элементам, поскольку планирование процессов, в ходе которого используется эта информация, не требуется. Аналогичным образом не нужно преобразовывать элементы конструкции в элементы изготовления. Достаточно иметь трехмерную поверхностную или твердотельную модель детали, на основе которой будут сгенерированы данные поперечных сечений.

Не требуется определять геометрию пустого пространства, поскольку в ходе процессов БПИ материал добавляется, а не снимается.

Не нужно определять несколько наборов оборудования или сложные последовательности обработки материала, поскольку деталь изготавливается за один прием.

Нет необходимости рассматривать конструкции зажимов и креплений. (Некоторые процессы могут требовать создания вместе с деталью поддерживающих структур. О поддерживающих структурах речь пойдет позже).

Не нужно проектировать и изготавливать формы и штампы, т. к. процессы БПИ являются безинструментальными.

Таким образом, поскольку процессы БПИ позволяют создать физический объект без использования инструментов, они хорошо подходят для интеграции проектирования и изготовления без планирования процессов.

Слои поперечных сечений могут создаваться и комбинироваться одним из следующих методов:

- полимеризация смол лазером, другими источниками света или лампами;
- избирательное спекание твердых частиц или порошка лучом лазера;
- связывание жидких или твердых частиц путем склеивания или сварки;

- резка и ламинирование листового материала;
- плавление и отверждение.

8.2. Процессы быстрого прототипирования и изготовления

В конце 70-х – начале 80-х гг. XX в. А. Герберт из корпорации 3М в Миннеаполисе, Х. Кодама из Исследовательского института префектуры Нагоя в Японии и К. Халл из корпорации Ultra Violet Products (UVP) в Калифорнии независимо друг от друга работали над идеями быстрого прототипирования, основанными на избирательном отверждении поверхностного слоя фотополимера и построении трехмерных объектов из последовательно наложенных слоев. Герберт и Кодама прекратили работу из-за недостатка финансирования, так и не сумев разработать коммерческий продукт. Халл, благодаря стабильной поддержке от UVP, разработал систему, способную автоматически изготавливать детали сложной формы. Халл ввел в обращение термин стереолитография (stereolithography) и в 1986 г. основал корпорацию 3D Systems, которая начала производить стереолитографические аппараты (stereo lithography apparatus – SLA).

Процесс изготовления детали изображен на рис. 8.1 и протекает следующим образом.

1. Фоточувствительный полимер, затвердевающий на свету, поддерживается в жидком состоянии.

2. На толщину одного слоя ниже поверхности жидкого полимера располагается платформа, способная двигаться в вертикальном направлении.

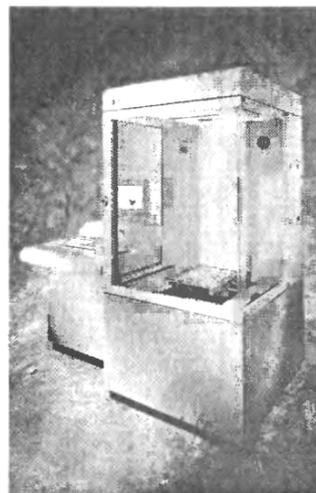
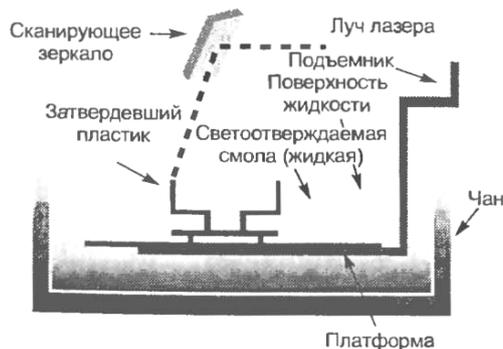
3. Ультрафиолетовый лазер сканирует слой полимера над платформой, отверждая полимер по форме соответствующего поперечного сечения. Процесс начинается с нижнего поперечного сечения детали.

4. Платформа опускается в ванну с полимером на толщину одного слоя, давая полимеру растечься по поверхности детали для начала нового слоя.

5. Шаги 3 и 4 повторяются, пока не будет наращен верхний слой детали.

6. Для полного затвердевания детали выполняется окончательное отверждение. Этот шаг необходим, поскольку в каждом слое могут еще оставаться жидкие участки. Так как лазерный луч имеет конечные размеры, сканирование каждого слоя аналогично закрашиванию некоторой фигуры тонкой цветной ручкой.

Реальный аппарат для стереолитографии показан на рис. 8.1, б.

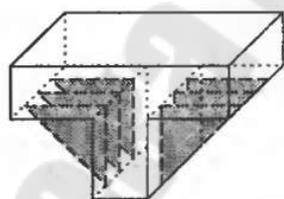


а)

б)

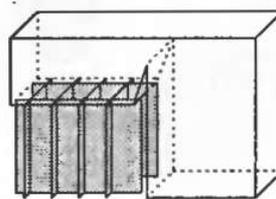
Рис. 8.1. Стереолитография:
а – процесс; б – аппарат SLA-3500 от 3D Systems

Стереолитография наиболее популярна среди процессов быстрого прототипирования и изготовления, и ее интерфейс с твердотельной моделью стал стандартом для других процессов. Однако она требует создания поддерживающих структур, если деталь имеет вырезы внизу, т. е. верхнее поперечное сечение детали имеет большую площадь, чем нижнее (рис. 8.2). Изготовленная методом стереолитографии крыльчатка изображена на рис. 8.2. Более подробно процесс стереолитографии будет описан далее (рис. 8.3).



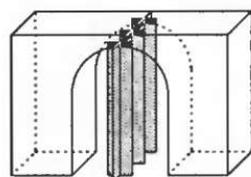
Угловое соединение

а)



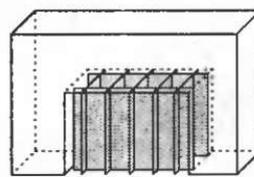
Островок

б)



Потолок с аркой

в)



Потолок

г)

Рис. 8.2. Поддерживающие структуры в стереолитографии

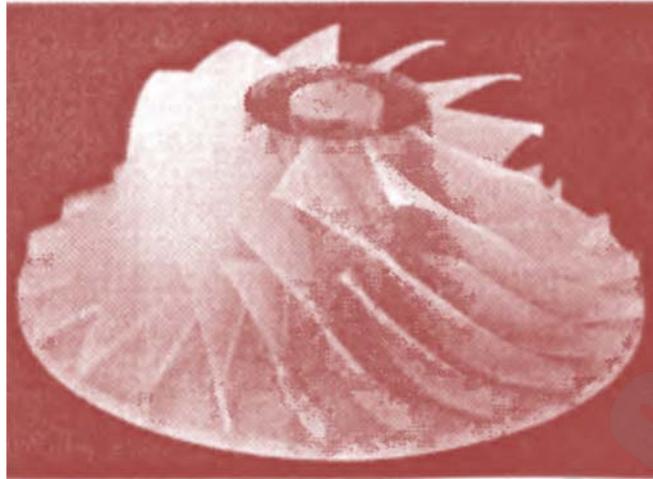


Рис. 8.3. Поддерживающие структуры в стереолитографии

Отверждение на твердом основании. В процессе отверждения на твердом основании (solid ground curing, SGC) каждый слой отверждается путем экспонирования ультрафиолетовой лампой, а не сканирования лазерным лучом. Таким образом, все точки слоя затвердевают одновременно и окончательное отверждение не требуется. Типичный процесс отверждения на твердом основании имеет место в системе bolider от Cubital Israel, работа которой происходит следующим образом.

1. По данным геометрической модели детали и желаемой толщине слоя рассчитывается поперечное сечение каждого слоя.

2. Для каждого слоя изготавливается оптическая маска по форме соответствующего поперечного сечения.

3. После выравнивания (рис. 8.4, а), платформа покрывается тонким слоем жидкого фотополимера (рис. 8.4, б).

4. Над поверхностью жидкой пластмассы помещается маска, соответствующая текущему слою, и пластмасса экспонируется светом мощной ультрафиолетовой лампы (рис. 8.4, в). Обратите внимание, что процесс начинается с маски, соответствующей нижнему слою.

5. Оставшаяся жидкость удаляется с изделия аэродинамическим wiper (рис. 8.4, г).

6. Изделие покрывается слоем жидкого воска, который заполняет пустоты (рис. 8.4, д). Затем к воску прикладывается холодная пластина и он затвердевает.

7. Слои стачиваются до желаемой толщины с помощью шлифовального диска (рис. 8.4, е).

8. Готовая часть изделия покрывается тонким слоем жидкого полимера, шаги 4–7 повторяются для каждого последующего слоя, пока не будет обработан самый верхний слой.

9. Воск расплавляется и удаляется из готовой детали.

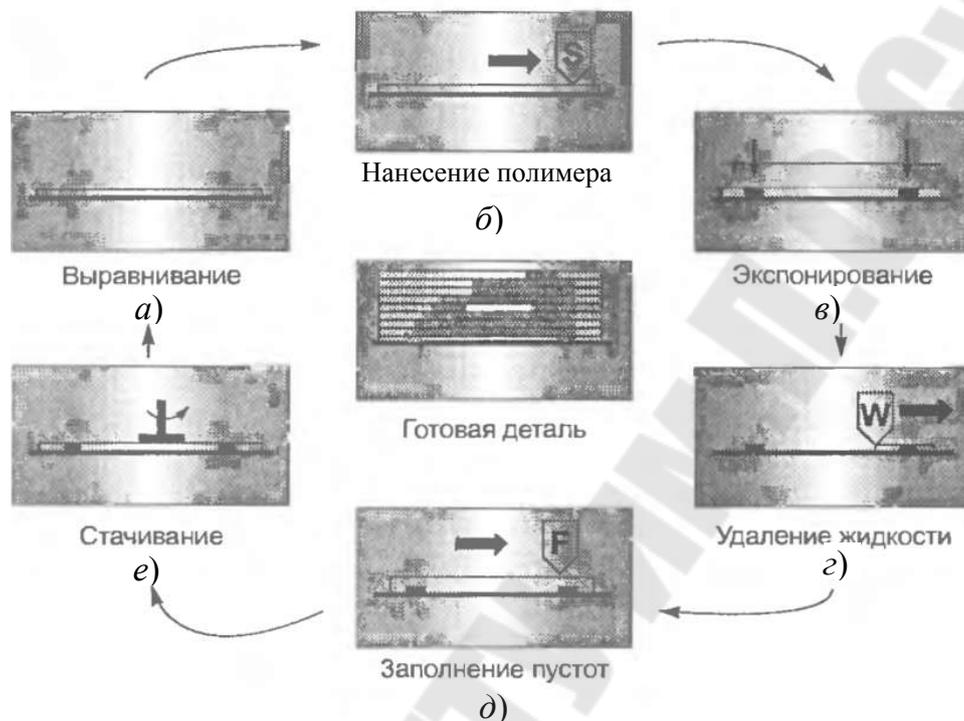


Рис. 8.4. Система Solider

Главным преимуществом отверждения на твердом основании по сравнению со стереолитографией является отсутствие необходимости в поддерживающих структурах. Это обусловлено тем, что пустоты заполняются воском. Кроме того, благодаря использованию света лампы вместо лазерных лучей исключается операция окончательного отверждения. Хотя отверждение на твердом основании позволяет изготавливать детали с большей точностью, чем стереолитография, процесс этот весьма сложен.

Избирательное лазерное спекание. Процесс изготовления детали путем избирательного лазерного спекания, разработанный фирмой DTM (США), протекает следующим образом.

1. Цилиндрическая заготовка помещается на высоте, необходимой для того, чтобы на нее можно было осадить слой порошкового материала желаемой толщины. Порошковый материал, используемый для изготовления прототипа, поступает из подающего цилиндра и наносится выравнивающим валиком (рис. 8.5).

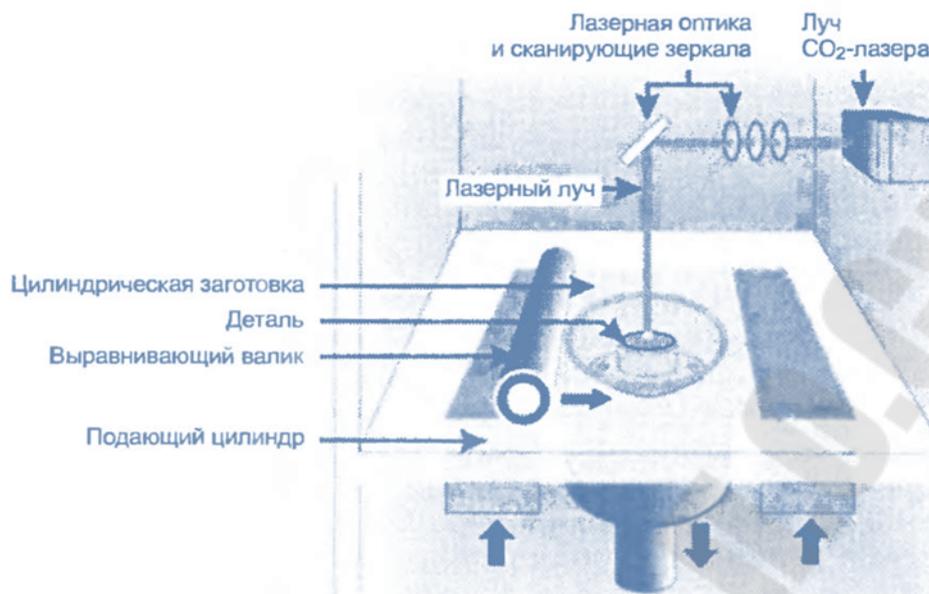


Рис. 8.5. Избирательное лазерное спекание

2. Слой порошка избирательно сканируется и нагревается лучом лазера, вследствие чего частицы слипаются между собой. Просканированные частицы порошка образуют требуемое поперечное сечение. Обратите внимание, что этот процесс начинается с нижнего поперечного сечения детали.

3. Цилиндрическая заготовка опускается на толщину одного слоя для нанесения нового слоя порошка.

4. Луч лазера сканирует новый слой порошка, склеивая его с предыдущим и формируя следующее поперечное сечение.

5. Шаги 3 и 4 повторяются, пока не будет создан самый верхний слой детали.

6. Для некоторых материалов может понадобиться окончательное отверждение.

Поддерживающая структура не требуется, потому что пустоты каждого слоя заполняются необработанным порошком. Более того, в качестве материала для процесса избирательного спекания потенциально может использоваться любой плавкий порошок, даже металлический, если лазер обладает достаточной мощностью. На практике для металлических порошков, частицы которых покрыты термопластическим связующим материалом, используется косвенное спекание. Под лучом лазера связующий материал расплавляется и свободно связывает частицы металлического порошка, образуя желаемую форму, которая называется «зеленой деталью» (green part). В этом случае достаточно, чтобы мощности лазера хватало для расплавления свя-

зующего материала. Затем зеленая часть подвергается обработке в печи, в ходе которой связующий материал выжигается, а частицы металлического порошка связываются за счет обычных механизмов спекания. Получившаяся деталь носит название «коричневой детали» (brown part). Без дальнейшей обработки деталь будет довольно пористой из-за наличия пустот, которые ранее занимали частицы связующего материала. Чтобы снизить пористость, в печь помещается еще один материал – инфильтрат. Этот металл расплавляется при рабочей температуре печи и проникает в поры детали за счет капиллярного эффекта. Данный метод используется для изготовления форм для литья непосредственно по их геометрическим моделям. Ресурса таких форм достаточно для изготовления от 2500 до 10 000 деталей-прототипов.

Трехмерная печать. Разработанный в Массачусетском Технологическом институте процесс трехмерной печати был назван так из-за своей схожести с печатью на струйном принтере. В трехмерной печати вместо чернил используется жидкое связующее вещество. Процесс трехмерной печати происходит следующим образом (рис. 8.6).

1. Платформа располагается на высоте, необходимой для того, чтобы можно было нанести на нее слой керамического порошка надлежащей толщины.

2. Нанесенный слой керамического порошка избирательно сканируется печатающей головкой, из которой поступает жидкое связующее вещество, вызывающее прилипание частиц друг к другу. Отсканированные печатающей головкой частицы образуют требуемую форму поперечного сечения. Обратите внимание, что этот шаг начинается с нижнего поперечного сечения.

3. Платформа опускается на одну толщину слоя, позволяя нанести следующий слой порошка.

4. Новый слой сканируется, образуя следующее поперечное сечение и склеиваясь с предыдущим слоем.

5. Шаги 3 и 4 повторяются, пока не будет создан верхний слой детали.

6. Для отверждения детали проводится последующая тепловая обработка.



Рис. 8.6. Трехмерная печать

С помощью трехмерной печати было бы удобно изготавливать формы для литья, поскольку форма изготавливается как цельная деталь, состоящая из оболочки и полостей, и положение полостей относительно оболочки можно было бы задавать точно. Однако формы для литья, изготовленные путем современной трехмерной печати, имеют невысокое качество поверхности.

Ламинирование. В процессе ламинирования (laminated object manufacturing – LOM), коммерциализированном фирмой Helisys, деталь изготавливается путем ламинирования и лазерной резки материалов, поступающих в листовом виде. Слипание листов происходит за счет наличия термоадгезивного покрытия. Процесс протекает следующим образом.

1. Каждый лист приклеивается к заготовке с помощью нагрева и давления, образуя очередной слой. Листовой материал подается в виде непрерывного рулона с одной стороны машины и принимается с противоположной стороны (рис. 8.7). Температуру и давление, необходимые для ламинирования, обеспечивает нагретый валик. Обратите внимание, что когда к стопке приклеивается следующий лист, платформа опускается на толщину одного листа.

2. После того как слой (лист) приклеен, он сканируется лазером вдоль контуров текущего поперечного сечения. Обычно для этой цели используется лазер на углекислом газе мощностью 25 или 50 Вт. Как и в других процессах, этот шаг начинается с нижнего поперечного сечения. Обратите внимание, что здесь сканирование производится

только по контурам. Это делает данный процесс более эффективным, чем процессы, требующие растрового сканирования.

3. Области слоя, выходящие за пределы контуров, штрихуются лазером (т. е. рассекаются на маленькие кусочки, называемые черепичками (tiles), для последующего удаления, когда деталь будет закончена).

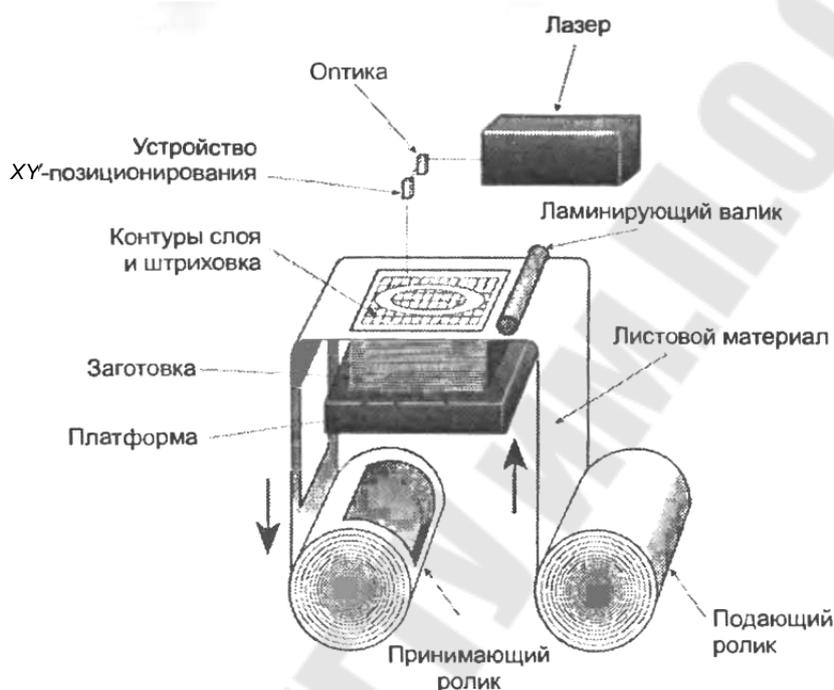


Рис. 8.7. Ламинирование

4. Шаги 1–3 повторяются до тех пор, пока не будет наклеен и вырезан верхний слой детали.

5. После того, как все слои будут готовы, результатом будет деталь, находящаяся внутри блока поддерживающего материала. Этот материал затем разламывается на кусочки вдоль линий лазерной штриховки.

6. Готовую деталь можно покрыть герметиком, чтобы предохранить ее от влажности.

Наличие поддерживающего материала вокруг детали имеет свои преимущества и недостатки. Прежде всего оно исключает необходимость во внешних поддерживающих структурах. При изготовлении детали внутри блока поддерживающего материала, имеющего определенную форму, геометрия всей структуры стабилизирована в процессе изготовления и, соответственно, ей не грозит перекося под собственным весом. Более того, нам не приходится беспокоиться об

изолированных «островках», которые часто образуются, когда твердое тело, спроектированное в САД-системе, пересекается на слои. Иными словами, ламинирование позволяет избежать создания специальных подпорок, которые точно фиксировали бы эти «островки» в пространстве, пока в процессе изготовления не будут созданы «мосты» к оставшимся частям детали. Однако удаление лишнего материала по окончании изготовления детали является непростой задачей.

Чтобы гарантировать, что удалены будут только излишки, а хрупкие части детали не будут при этом сломаны, необходима бережная очистка, выполняемая вручную. Кроме того, полую структуру с замкнутыми поверхностями невозможно изготовить в виде единой части, поскольку в этом случае излишки материала невозможно будет извлечь изнутри. Сложность удаления ненужного материала характеризует любую часть с узкими перемычками, внутренними полостями с ограниченным доступом, слепыми отверстиями и т. п. Далее большая часть материала, расходуемая при ламинировании, идет не на саму деталь, а остается неиспользованной в рулоне или образует поддерживающие структуры, которые будут удалены после изготовления. Это может быть весьма расточительно, если применяются более дорогостоящие материалы, чем бумага.

Это субтрактивный, а не аддитивный процесс (т. е. для создания слоя с требуемым поперечным сечением материал удаляется, а не добавляется). Во всех прочих процессах БПИ слои образуются путем добавления материала. Таким образом, потенциально ламинирование является самой быстрой технологией изготовления деталей с большим отношением объема к площади поверхности. Детали образуются перемежающимися слоями материала и клейкого вещества. Соответственно, многие из их физических свойств являются неоднородными и анизотропными. Потенциальная точность процесса изготовления ламинированных объектов высока. В нем может использоваться сколь угодно тонкий листовой материал, что позволяет достичь хорошей разрешающей способности в направлении наращивания детали. В действительности изготовить тонкий однородный листовой материал несложно, и усадка при ламинировании не представляет проблемы, поскольку контуры вырезаются после того, как происходит усадка. Хотя процесс потенциально применим ко многим материалам, включая пластики, композиты и металлы, наиболее популярным на сегодняшний день является бумажное ламинирование.

Моделирование методом наплавления. В процессе наплавления (fused-deposition modeling – FDM), коммерциализированном фир-

мой Stratasys, каждый слой формируется путем выдавливания термопластичного материала, находящегося в жидком состоянии (рис. 8.8).

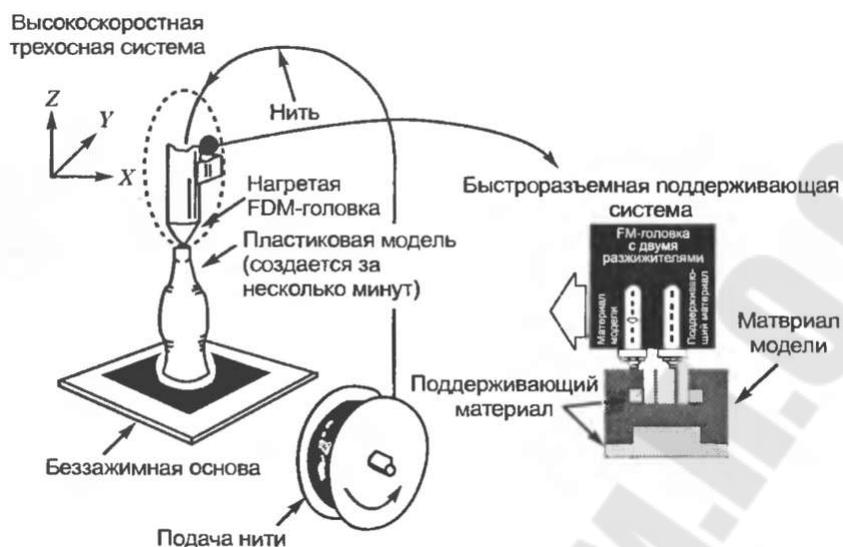


Рис. 8.8. Моделирование методом наплавления

Температура выдавливаемого материала незначительно превышает его температуру затвердевания: это аналогично созданию надписей на торте шоколадным кремом. Деталь изготавливается путем последовательного наплавления слоев. Этот процесс относительно прост, но его применение ограничено термопластичными материалами.

Литература

1. Федорук, В. Г. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 3. Информационное и прикладное программное обеспечение / В. Г. Федорук, В. М. Черненко ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1988. – 156 с.
2. Трудоношин, В. А. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 4. Математические модели технических объектов / В. А. Трудоношин, Н. В. Пивоваров ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1988. – 160 с.
3. Кузьмик, П. К. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 5. Автоматизация функционального проектирования / П. К. Кузьмик, В. Б. Маничев ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1988. – 142 с.
4. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 8. Сборник примеров и задач / Д. М. Жук [и др.] ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1988. – 141 с.
5. Имитационное моделирование в задачах синтеза структуры сложных систем: Оптимизационно-имитационный подход / А. Д. Цвиркун [и др.]. – Москва : Наука, 1985. – 173 с.
6. Бешелев, С. Д. Математико-статистические методы экспертных оценок / С. Д. Бешелев, Ф. Г. Гурвич. – Москва : Статистика, 1980. – 263 с.
7. Норенков, И. П. Введение в автоматизированное проектирование технических устройств и систем : учеб. пособие для вузов / И. П. Норенков. – 2-е изд., перераб. и доп. – Москва : Высш. шк., 1986. – 304 с.
8. Норенков, И. П. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 1. Принципы построения и структура / И. П. Норенков ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1987. – 123 с.
9. Жук, Д. М. САПР. Системы автоматизированного проектирования : учеб. пособие для вузов. В 9 кн. Кн. 2. Технические средства и операционные системы / Д. М. Жук, В. А. Мартынюк, П. А. Сомов ; под ред. И. П. Норенкова. – Минск : Выш. шк., 1988. – 155 с.
10. Челищев, Б. Е. Автоматизация проектирования технологии в машиностроении / Б. Е. Челищев, И. В. Боброва, А. Гонсалес-Сабатер. – Москва : Машиностроение, 1987. – 264 с.
11. Любарский, Ю. Я. Интеллектуальные информационные системы / Ю. Я. Любарский. – Москва : Наука, 1990. – 232 с.

12. Львов, Б. Г. Основы теории технических систем / Б. Г. Львов. – Москва : МИЭМ, 1991. – 136 с.

13. Солодовников, И. В. Языки, программное обеспечение и организация систем имитационного моделирования / И. В. Солодовников. – Москва : Машиностроение, 1982. – 48 с.

14. Шпур, Г. Автоматизированное проектирование в машиностроении / Г. Шпур, Ф. Л. Краузе. – Москва : Машиностроение, 1988. – 648 с.

15. Першиков, В. И. Толковый словарь по информатике / В. И. Першиков, В. М. Савинков. – Москва : Финансы и статистика, 1991. – 543 с.

16. Ткаченко, А. В. Система автоматизированного проектирования технологических процессов, оснастки и оборудования : практ. пособие по выполнению лаборатор. работ по одноим. курсу для студентов специальности 1-36 02 01 «Машины и технология литейного производства» днев. и заоч. форм обучения / А. В. Ткаченко. – Гомель : ГГТУ им. П. О. Сухого, 2006. – 54 с.

17. Ткаченко, А. В. Системы автоматизированного проектирования технологических процессов, оснастки и оборудования : метод. указания к контрол. работам по одноим. курсу для студентов специальности 1-36 02 01 «Машины и технология литейного производства» заоч. формы обучения / А. В. Ткаченко. – Гомель : ГГТУ им. П. О. Сухого, 2007. – 14 с.

18. Ткаченко, А. В. Системы автоматизированного проектирования технологических процессов, оснастки и оборудования : метод. указания к курсовой работе для студентов специальности 1-36 02 01 «Машины и технология литейного производства» заоч. формы обучения / А. В. Ткаченко. – Гомель : ГГТУ им. П. О. Сухого, 2007. – 26 с.

Содержание

Введение.....	3
1. Основы систем автоматизированного проектирования.	
Общие положения. Компоненты систем автоматизированного проектирования	5
1.1. Введение в проектирование с использованием САПР	5
1.2. Определение CAD, CAM, CAE.....	13
1.3. Структура и параметры объектов проектирования	17
1.4. САПР на базе Windows. Объектно-ориентированное программирование в САПР	18
2. Системы автоматизированного проектирования технологических процессов литейного производства	21
2.1. Автоматизация процесса разработки технологии получения отливок.....	21
2.2. Применение систем автоматизированного моделирования литейных процессов в опытном производстве литых заготовок....	26
2.3. Структура отдела САПР литейного производства	33
3. Обработка данных в системе автоматизированного проектирования	35
3.1. Стандарты обмена данными между системами	35
3.2. Методы обмена данными технических требований	35
3.3. Форматы файлов обмена данными в САПР	37
3.3.1. Формат IGES	37
3.3.2. Формат DXF	40
3.3.3. Формат STEP	41
4. Основные концепции графического программирования.	
Системы автоматизированной разработки чертежей.....	45
4.1. Основные концепции графического программирования	45
4.1.1. Системы координат.....	48
4.1.2. Окно и видовой экран.....	52
4.2. Примитивы в графических построениях	54
4.3. Системы автоматизированной разработки чертежей.....	59
5. Системы геометрического моделирования	66
5.1. Системы каркасного моделирования.....	66
5.2. Системы поверхностного моделирования.....	67
5.3. Системы твердотельного моделирования	69

5.3.1. Функции моделирования	70
5.3.2. Объектно-ориентированное моделирование	73
5.3.3. Параметрическое моделирование	74
6. PLM-технологии. Интеграция систем автоматизированного проектирования	76
6.1. Особенности процесса синтеза программных систем.....	79
7. Метод конечных элементов в системе автоматизированного проектирования	84
7.1. Введение в метод конечных элементов	84
7.2. Решение задач проектирования с использованием программного пакета «ANSYS».....	87
8. Быстрое прототипирование и изготовление	110
8.1. Обзор	110
8.2. Процессы быстрого прототипирования и изготовления.....	112
Литература	122

Учебное электронное издание комбинированного распространения

Учебное издание

Ткаченко Александр Владимирович
Жаранов Виталий Александрович

**САПР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ,
ОСНАСТКИ И ОБОРУДОВАНИЯ.
САПР МЕТАЛЛУРГИЧЕСКОГО
ПРОИЗВОДСТВА**

Курс лекций
по одноименной дисциплине
для студентов специальностей
1-36 02 01 «Машины и технология литейного
производства» и 1-42 01 01 «Металлургическое
производство и материалобработка»
дневной и заочной форм обучения

Электронный аналог печатного издания

Редактор *Н. И. Жукова*
Компьютерная верстка *Н. Б. Козловская*

Подписано в печать 15.02.11.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 7,44. Уч.-изд. л. 7,39.

Изд. № 61.

E-mail: ic@gstu.by

<http://www.gstu.by>

Издатель и полиграфическое исполнение:
Издательский центр учреждения образования
«Гомельский государственный технический университет
имени П. О. Сухого».

ЛИ № 02330/0549424 от 08.04.2009 г.

246746, г. Гомель, пр. Октября, 48.